ADJOINT-BASED AEROELASTIC OPTIMIZATION WITH HIGH-FIDELITY TIME-ACCURATE ANALYSIS

A Dissertation
Presented to
The Academic Faculty

By

Kevin E. Jacobson

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology

May 2019

ADJOINT-BASED AEROELASTIC OPTIMIZATION WITH HIGH-FIDELITY TIME-ACCURATE ANALYSIS

Approved by:

Dr. Marilyn J. Smith, Co-Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Graeme J. Kennedy, Co-Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Stephen M. Ruffin
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Brian J. German
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Steven J. Massey
Aeroelasticity Branch
NASA Langley Research Center

Mr. William T. Jones
Computational Aerosciences Branch
NASA Langley Research Center

Date Approved: November 27, 2018

ACKNOWLEDGEMENTS

First, I would like to express my gratitude to my advisors at Georgia Tech, Dr. Marilyn Smith and Dr. Graeme Kennedy. This work has been greatly impacted by their guidance and would not have been possible without their support. I would also like to thank my Ph.D. committee, especially for their patience as the subject of my dissertation shifted. Steve Massey and Bill Jones have been very supportive as I have transitioned from Georgia Tech to NASA, and a lot of the coupling interface design is based on "Bill's Grand Vision" and his other suggestions.

Thank you to the Aeroelasticity and Computational AeroSciences Branches at NASA Langley for sharing their wisdom with me. I am very grateful to Pawel Chwalowski, Bret Stanford, and Jen Heeg for providing input decks and data, assisting with post-processing, and letting me bounce ideas, questions, and results off of them. I would also like to thank Boris Diskin and Li Wang from the National Institute of Aerospace for their help with mesh movement adjoint term derivations.

Thank you to all of the members of the Nonlinear Computational Aeroelasticity Lab and the Structures and Multidisciplinary Design Optimization Lab at Georgia Tech for their support, especially, Ting Wei Chin and Komahan Boopathy, who helped me with TACS; Amanda Grubb, who had the answer to all my CFD meshing troubles; and Jan Kiviaho. As the other graduate student on the project that funded this work, Jan implemented some of the key components, and I could not have asked for a better person to work with.

Finally, thank you to my family and friends for their unwavering support during my time in graduate school.

Funding for this work was provided through the NASA Transformative Tools and Technologies program with grant number NNX15AU22A with Technical Monitor Steve Massey. Computational resources were provided by the NASA High-End Computing Program through the NASA Advanced Supercomputing Division at Ames Research Center.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

$\mathbf{A}$	Aerodynamic residual
$\mathbf{A}_\psi$	Aerodynamic adjoint residual
b	Semichord
c	Prony series complex amplitude
$\mathbf{C}$	Damping matrix
C_D	Drag coefficient
C_L	Lift coefficient
C_{My}	Moment coefficient
$\mathbf{D}$	Displacement transfer residual
$\mathbf{D}_\psi$	Displacement transfer adjoint residual
$\mathbf{E}$	Local elastic deformation residual
$\mathbf{E}_\psi$	Local elastic deformation adjoint residual
f	Function of interest
$\mathbf{f}_A$	Aerodynamic surface forces
$\mathbf{f}_S$	Structural forces
$\mathbf{F}_\psi$	Force integration adjoint residual
$\overline{\overline{\mathbf{F}}}^*$	Convective flux term
$\overline{\overline{\mathbf{F}}}_\mathbf{v}$	Diffusive flux (viscous and turbulence model)
$\mathbf{G}$	CFD volume grid deformation residual
$\mathbf{G}_\psi$	Volume grid deformation adjoint residual
h	Amplitude, perturbation size
$\mathbf{K}$	Structural stiffness matrix

$\mathbf{K}_G$	CFD grid deformation stiffness matrix
L	Pencil parameter
$\mathbf{L}$	Load transfer residual
$\mathbf{L}_\psi$	Load transfer adjoint residual
$\mathbf{M}$	Structural mass matrix
M	Model order
$\hat{\mathbf{n}}$	Unit normal vector
$\mathbf{q}$	Aerodynamic state vector
$\mathbf{R}$	Rigid motion extraction residual
$\mathbf{R}_\psi$	Rigid motion extraction adjoint residual
s	Prony series complex exponent
$\mathbf{S}$	Structural residual
$\mathbf{S}_\psi$	Structural adjoint residual
t	Time
$\mathbf{T}$	Rigid transformation matrix
$\mathbf{U}$, $\mathbf{V}$	Left and right singular vectors
$\mathbf{u}_A$	Surface mesh displacement vector
$\mathbf{u}_{A*}$	Surface mesh displacement vector, transformed frame
$\mathbf{u}_S$	Structural displacements
V	Control volume
$\mathbf{V}$	Volume node displacement vector
$w(n)$	Matrix pencil noise term
$\mathbf{x}$	Design variable vector
$\mathbf{x}_{A0}$	Surface mesh coordinates after shape parameterization changes
$\hat{\mathbf{x}}_A$	Initial aerodynamic surface coordinates
x_c	Coordinate of the center of mass measured from elastic axis
x_{EA}	Coordinate of the elastic axis measured from the midchord

$\mathbf{x}_G$	Deformed volume grid coordinates
$\hat{\mathbf{x}}_G$	Initial volume grid coordinates
$\mathbf{x}_{S0}$	Structural mesh coordinates after shape parameterization changes
$\mathbf{Y}$	Hankel Matrix
z	Displacement
z_v	Hilbert transform of z
α	Exponential growth coefficient
β	Transfer scheme weighting factor
ζ	Damping ratio
η	Action of the load transfer
θ	Aitken under-relaxation factor
λ	Eigenvalue vector
ν	Poisson's ratio
ξ	Action of the displacement transfer
Σ	Singular value vector
ϕ	Action of the force integration
Ψ, Φ	Left and right eigenvectors
ψ_A	Aerodynamic adjoint vector
ψ_D	Displacement transfer adjoint vector
ψ_E	Local elastic deformation adjoint vector
ψ_F	Force integration adjoint vector
ψ_G	Volume grid deformation adjoint vector
ψ_L	Load transfer adjoint vector
ψ_R	Rigid motion extraction adjoint vector
ψ_S	Structural adjoint vector
ω	Frequency

SUMMARY

The design of traditional tube and wing aircraft relies heavily on tools based on linear physics and empirical models. However, as engineers are pursuing aggressive performance goals and exploring nontraditional aircraft configurations, the applicability of traditional design tools is diminishing. These new design problems involve nonlinear physics related to aeroelasticity. As algorithms and computer hardware advance, the use of high-fidelity tools, which employ computational fluid dynamics and finite element modeling, is becoming a more realistic possibility for aircraft design. The incorporation of high-fidelity design tools earlier in the design process will allow the nonlinear physics to be more accurately modeled. More accurate analysis in the design process will lead to more confidence in designs and fewer costly late design cycle modifications.

One approach to introduce high-fidelity optimization tools in a computationally efficient manner is the adjoint method. The adjoint method has been applied to high-fidelity aeroelastic design problems, but the current literature typically focuses on certain aspects of the design problem while making simplifying assumptions or ignoring other aspects of the aeroelastic problem. For example, some state-of-the-art design efforts ignore unsteady aeroelastic effects, assume small deflections, or do not consider structural design aspects of the problem. This dissertation presents a more general aeroelastic formulation that allows the implementation of aerodynamic, structural, and aeroelastic design constraints. The aeroelastic coupling includes steady and time-accurate analysis with corresponding adjoint formulations. The aerodynamic mesh movement in the formulation can decompose the motion into rigid and deformation components which can capture large motion.

In addition to the coupling formulation, some unsteady aeroelastic constraint formulations are presented. Flutter constraints based on the matrix pencil method are demonstrated for the identification of flutter conditions. Aggregated stress constraints for gust responses are illustrated where the gust is modeled with the field velocity method.

CHAPTER 1
INTRODUCTION

1.1 Motivation

Aeroelasticity plays an important role in aerospace design. Certification of fixed wing aircraft requires sufficient gust and flutter margins throughout the flight envelope. Likewise, turbomachinery, rotorcraft, wind turbines, and launch vehicles are all susceptible to a range of aeroelastic problems including divergence, flutter, limit cycle oscillations, control reversal, and buffet. These problems can involve nonlinear physics due to structural stiffness or damping nonlinearity, geometric nonlinearity, or aerodynamic nonlinearity such as shock movement or flow separation [1]. The complexity of these problems is increasing as advanced materials permit engineers to improve performance with thinner and more flexible designs where nonlinearities can exist simultaneously in the structures and aerodynamics. Aeroelastic fixed wing aircraft design is predominately conducted with linear aerodynamic tools followed by wind tunnel and flight tests [2]. However, lower-order and linear tools are insufficient to predict complicated aeroelastic behaviors such as the transonic flutter dip. This leads to heavy, overly conservative structural design or the need to modify configurations when problems arise late in the design process. Advancements in solution algorithms and more powerful computers permit high fidelity multidisciplinary modeling based on finite element analysis and computational fluid dynamics (CFD) to be more tractable and will push these tools to be accessible earlier in the design cycle. If more accurate models for the complex aeroelastic phenomena can be included earlier in the design process, aircraft designers will have more confidence in their designs before flight tests, and the need for overly conservative designs or costly late-stage modifications can be reduced. In addition to avoidance of aeroelastic problems, including aeroelasticity in the optimization process

can lead to designs to utilize aeroelasticity in a beneficial way. For example, aeroelastic washout during maneuvers can relieve stress at the wing root by shifting the lift distribution inboard.

This need for high-fidelity multidisciplinary analysis and optimization (MDAO) is emphasized in the CFD Vision 2030 study [3]. The CFD 2030 study is a forecast and recommended strategy for National Aeronautics and Space Administration (NASA) for what CFD capability the organization should have by the year 2030. In the report the authors provide four grand challenge problems. One of the challenges is multidisciplinary analysis and optimization of a highly flexible advanced aircraft configuration including "explicit aeroelastic constraints that may require a time-accurate CFD approach." The report states that by 2030, "multidisciplinary simulations will become the norm rather than the exception" and "CFD will have to interface seamlessly with other high-fidelity analyses including acoustics, structures, heat transfer, reacting flows, radiation, dynamics and controls, ..." The report notes that one of the impediments to enabling multidisciplinary analysis and optimization by 2030 is lack of sensitivity information for optimization and uncertainty quantification. This dissertation represents a step towards the 2030 goals by developing the capability to calculate sensitivities efficiently for time-accurate high-fidelity analysis of aeroelastic systems with a modular, extensible, and efficient computational framework.

1.2 Optimization Algorithms

Optimization algorithms are usually classified as either gradient-free or gradient-based methods [4]. Gradient-based methods utilize the sensitivities of the objectives and constraints to design variables to determine where to move in the design space to minimize the objective while satisfying the constraints. Gradient-based optimization methods find a local optimum, but compared to gradient-free methods, they can scale much move favorably with respect to the size of the design problem.

As the name suggests, gradient-free methods do not require sensitivity information to

find the optimum solution. Some of these methods such as genetic algorithms [5–7], particle swarm optimization [8], or simulated annealing [9, 10] are inspired by natural processes. Others such as Bayesian optimization [11] or random search methods [12, 13] have more mathematical or statistical roots. Gradient-free methods have advantages as they search for global optimums, and they can handle discrete design variables and non-continuous or non-differentiable functions. However, they do not scale as efficiently with number of design variables as gradient-based methods.

Gradient-based methods can be further classified as either line search methods [4, 14, 15] or trust-region methods [16, 17]. A generic line search method is given in Algorithm 1. Line search methods work by selecting a direction in the design space, approximately solving the one-dimensional minimization problem along that direction, updating the search direction, and repeating until a local minimum is found. Various line search algorithms have different methods for determining the search direction and initial step sizes for the one-dimensional minimization. They can also have different conditions for satisfying the approximate one-dimensional minimization problem.

Algorithm 1 Generic line search method.

Given: $\beta \in (0,1), \gamma \in (0,1)$,$\mathbf{x}_0$

$f(\mathbf{x}_0)$, $\nabla f(\mathbf{x}_0)$ ▷ Evaluate the function and gradient

$\mathbf{p}_0 \leftarrow \nabla f(\mathbf{x}_0)$ ▷ Set the initial search direction

for $n = 1, n_{max}$ **do** ▷ Optimization loop

  if $\|\nabla f(\mathbf{x}_n)\| < tol$ **then** ▷ Local minimum check

    Break optimization loop

  end if

  $\alpha_0 \leftarrow \alpha(f(\mathbf{x}_{n-1}), \nabla f(\mathbf{x}_{n-1}), ...)$ ▷ Determine the initial step size

  m = 0

  while True **do** ▷ Line search loop

    if $f(\mathbf{x}_n + \alpha_m \mathbf{p}_n) < f(\mathbf{x}_n) + \beta \alpha_m \nabla f(\mathbf{x}_n)^T \mathbf{p}_n$ **then** ▷ Line search convergence

      $\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_m \mathbf{p}_n$ ▷ Update the position

      Break line search loop

    else

      $\alpha_{m+1} = \gamma \alpha_m$ ▷ Reduce the step size

      $m = m + 1$

    end if

  end while

  $\mathbf{p}_{n+1} \leftarrow \mathbf{p}(f(\mathbf{x}_{n+1}), \nabla f(\mathbf{x}_{n+1}), f(\mathbf{x}_n), ...)$ ▷ Update the search direction

end for

A generic trust-region method is given in Algorithm 2. In trust-region methods, an approximate model is formed about the current design point, and the minimum is found on this approximate model. Because this model is approximate, it will lose accuracy as the distance from the current design point is increased; therefore, the minimization on the approximate model is restricted to a region of trust around the current design point. After determining the approximate minimizer, the true function value is computed at that point.

The algorithm can accept that point then repeat the process, or it can reject that point and adjust the trust region size. Trust-region methods have different ways of forming the approximate model and trust region's shape and size selection.

Algorithm 2 Generic trust-region method.

Given: $\Delta_0, \Delta_{max}, \eta \in [0, 1/4), \mathbf{x}_0$

for $n = 0, n_{max}$ **do** ▷ Optimization loop

  if $\|\nabla f(\mathbf{x}_n)\| < tol$ **then** ▷ Local minimum check

    Break optimization loop

  end if

  $m_k(\mathbf{x}_n + \mathbf{p}) \leftarrow m(f(\mathbf{x}_n), \nabla f(\mathbf{x}_n, ...))$ ▷ Update the approx. model

  $\mathbf{p}_k \leftarrow \min m_k(\mathbf{x}_k + \mathbf{p}),\ \|\mathbf{p}\|_2 < \Delta_n$ ▷ Approx. model min. within the trust region

  $\rho_n = \dfrac{f(\mathbf{x}_n) - f(\mathbf{x}_n + \mathbf{p}_n)}{m_n(\mathbf{x}_n) - m_n(\mathbf{x}_n + \mathbf{p}_n)}$ ▷ Measure the accuracy of the approx. model

  if $\rho_n \geq \eta$ **then** ▷ Update the position

    $\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{p}_n$

  else

    $\mathbf{x}_{n+1} = \mathbf{x}_n$

  end if

  if $\rho_n < 0.25$ **then** ▷ Update the trust region radius

    $\Delta_{n+1} = 0.25\Delta_n$

  else if $\rho_n > 0.75$ and $\|\mathbf{p}_n\|_2 = \Delta_n$ **then**

    $\Delta_{n+1} = \min(2\Delta_n, \Delta_{max})$

  else

    $\Delta_{n+1} = \Delta_n$

  end if

end for

The ideal choice of optimization algorithm is problem dependent. It depends on factors such as the number of design variables, the number of objective functions and constraints,

properties of the design space (size, smoothness, flatness, etc.), the cost of each function evaluation, and whether a local optimum is sufficient or a global optimum is necessary. Lyu et al. [18] compared the number of function evaluations required for gradient-based and gradient-free optimization algorithms in the context of CFD-based optimization. The authors compared the algorithms for a range of design variables on a multi-dimensional Rosenbrock function [19]. The gradient-free methods scaled approximately quadratically with the number of design variables, while the gradient-based methods scaled linearly or better than linearly. With the SNOPT gradient-based optimizer [20] and analytical gradient calculation, the number of function evaluations required to solve the multi-dimensional Rosenbrock function was essentially independent of the number of design variables.

To have a rich design space, aerospace design problems require thousands or more design variables. For the design of a wing, design variables can include parameters that control the outer mold line shape; skin thickness; composite layup angles; or the location and size of internal structural components, engines, or control surfaces. Assuming thousands of design variables, the millions of high-fidelity simulations required to find a global optimum with gradient-free methods is an intractable problem with the current status of high performance computing and numerical algorithms. To keep the problem size reasonable, a more appropriate process would be to utilize lower fidelity tools to explore the design space more fully with a gradient-free method (conceptual design), select one or more configurations from this process which are potential candidates to be the global optimum, and then transition to high-fidelity gradient-based methods to optimize with more accurate modeling of the physics (preliminary and/or detailed design). As discussed in Section 1.5, almost all of CFD-based optimizations follow a similar approach and use a gradient-based method when CFD is involved.

1.3 Methods for Computing Gradients

Within the realm of gradient-based methods, there are multiple ways to compute sensitivities. The simplest method is real-valued finite differences with small perturbations. When applied with finite-precision arithmetic, real-valued finite differences are subject to subtractive cancellation which can lead to inaccurate approximations of the gradient. Therefore, a balance must be found between truncation error due to too large of a step size and subtractive cancellation due to too small of a step size. Another disadvantage of the real-valued finite-difference approach is that the number of function evaluations per design point is proportional to the number of design variables. For these reasons, the real-valued finite-difference approach is not a common method in the literature except to verify derivatives calculated from the other methods.

Complex step finite differences for real-valued functions is a more accurate alternative to real-valued finite differences [21–23]. A complex number perturbation is applied to a input variable, and the sensitivity of the function to that variable is given by

$$\frac{\partial f(x)}{\partial x} = \frac{\mathrm{Im}(f(x+ih))}{h} + O(h^2). \tag{1.1}$$

This expression is derived from a Taylor Series centered about $f(x)$ with a step size of ih. Unlike real-valued finite differences, this expression does not include subtraction. Therefore, it is not subject to subtractive cancellation and can be applied with extremely small step sizes (10^{-30}-10^{-50}) which can compute sensitivities to machine precision. Like the real-valued finite-difference method, the complex step approach does not required solving additional equations to obtain the sensitivities which makes it simple to implement. But there is additional computational cost associated with replacing all of the real-valued operations with complex versions. Each complex solver evaluation is approximately three times the cost of the original solver [24]. The first application of the complex step method to aeroelastic problems by Newman et al. [24] who demonstrated the superior accuracy of

the complex-valued finite difference sensitivities compared to real-valued finite differences for an ONERA M6 wing coupled to a finite element structure representing seven ribs and two spars. Like the real-valued finite difference method, the sensitivity of any number of functions can be obtained with a single perturbation; however, the number of simulations required to form the full gradient scales linearly with the number of design variables. This lack of scalability has limited the complex step method's application in high-fidelity optimization. However, because it is one of the simpler sensitivity methods to implement and can be accurate to machine precision, it has been as widely adopted for verification of implementation of other methods of calculating sensitivities.

The tangent and adjoint methods for computing gradients are closely related methods where sensitivities are determined by solving an additional linear problem [25]. Given a function $f(\mathbf{x},\mathbf{q})$ and a set of governing equations $\mathbf{R}(\mathbf{x},\mathbf{q})=0$ where $\mathbf{x}$ is the design variable vector and $\mathbf{q}$ is the state vector, the derivative of the function with respect to the design variables is

$$\frac{df}{d\mathbf{x}}=\frac{\partial f}{\partial \mathbf{x}}+\frac{\partial f}{\partial \mathbf{q}}\frac{\partial \mathbf{q}}{\partial \mathbf{x}}. \tag{1.2}$$

The relationship between the state and design variables is governed by the residual, $\mathbf{R}(\mathbf{x},\mathbf{q})=0$. Differentiating this residual results in

$$\frac{d\mathbf{R}}{d\mathbf{x}}=\frac{\partial \mathbf{R}}{\partial \mathbf{x}}+\left[\frac{\partial \mathbf{R}}{\partial \mathbf{q}}\right]\frac{\partial \mathbf{q}}{\partial \mathbf{x}}=0, \tag{1.3}$$

and rearranging leads to

$$\frac{\partial \mathbf{q}}{\partial \mathbf{x}}=-\left[\frac{\partial \mathbf{R}}{\partial \mathbf{q}}\right]^{-1}\frac{\partial \mathbf{R}}{\partial \mathbf{x}}. \tag{1.4}$$

This expression can be substituted into Equation 1.2

$$\frac{df}{d\mathbf{x}}=\frac{\partial f}{\partial \mathbf{x}}-\frac{\partial f}{\partial \mathbf{q}}\left[\frac{\partial \mathbf{R}}{\partial \mathbf{q}}\right]^{-1}\frac{\partial \mathbf{R}}{\partial \mathbf{x}}. \tag{1.5}$$

At this point, the tangent and adjoint methods diverge. In the tangent method which is also

referred as the direct method, the final two components of the second term are combined into a variable ϕ,

$$\phi = -\left[\frac{\partial \mathbf{R}}{\partial \mathbf{q}}\right]^{-1} \frac{\partial \mathbf{R}}{\partial \mathbf{x}}, \tag{1.6}$$

or

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{q}}\right] \phi = -\frac{\partial \mathbf{R}}{\partial \mathbf{x}}. \tag{1.7}$$

The value of ϕ can be found by solving this linear problem, known as the tangent problem. Once ϕ is found, the sensitivity is given by

$$\frac{df}{d\mathbf{x}} = \frac{\partial f}{\partial \mathbf{x}} + \frac{\partial f}{\partial \mathbf{q}} \phi \tag{1.8}$$

The tangent problem in Equation 1.7 is independent of the number of functions being evaluated, but it is dependent on the design variables. The only calculation dependent on the number of functions is in the matrix-vector product in Equation 1.8. Compared to a high fidelity analysis, this final matrix-vector product is computationally inexpensive, so the tangent method is essentially independent of the number of functions of interest but requires one tangent problem solution per design variable.

For the adjoint method, the first two components of the second term in Equation 1.5 are combined into a new variable ψ

$$\psi^T = -\frac{\partial f}{\partial \mathbf{q}} \left[\frac{\partial \mathbf{R}}{\partial \mathbf{q}}\right]^{-1}, \tag{1.9}$$

which can be rearranged as

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{q}}\right]^T \psi = -\left[\frac{\partial f}{\partial \mathbf{q}}\right]^T. \tag{1.10}$$

This linear equation set is known as the adjoint problem. The adjoint problem is independent of the design variables, $\mathbf{x}$, but the function of interest appears on the right hand side of the equation. After solving the adjoint problem for the adjoint vector, ψ, the sensitivity

expression in Equation 1.5 becomes

$$\frac{df}{d\mathbf{x}} = \frac{\partial f}{\partial \mathbf{x}} + \psi^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}}. \tag{1.11}$$

Like the tangent method, the final gradient assembly is inexpensive compared to solving the forward and adjoint problem. The only time that the design variables appear in the process is Equation 1.11. Therefore, the adjoint method is essentially independent of the number of design variables, but one adjoint problem must be solved for each function of interest. The adjoint method is preferred over the tangent method when the number of objectives and constraints is smaller than the number of design variables, which is typically the case for aeroelastic design. Since its introduction from control design to CFD by Jameson [25], the adjoint method has been applied to wide range of design problems from steady airfoil and wing design [26] to sonic boom reduction [27] and unsteady aeroacoustics optimization [28].

There are two kinds of adjoint formulations: the continuous adjoint and the discrete adjoint. In the discrete adjoint, the adjoint equations are derived from the discretized version of the governing equations. For the continuous adjoint, the adjoint method is applied to the exact partial differential equation, and the resulting adjoint equations are then discretized. Both adjoint methods have proven to be useful for gradient-based optimization, and both methods have their own advantages. If great care is take in discretization and implementation of the continuous adjoint, the two methods can produce exactly the same derivatives values. Such methods are called dual consistent [29, 30]. Peter and Dwight describe the differences between the discrete and continuous methods in Reference [31]. Peter and Dwight state that the continuous adjoint is often less expensive in terms of operation counts and memory, although this can be implementation dependent. The discrete adjoint will always compute the exact gradient of the discrete function of interest, whereas for the continuous adjoint this is only possible (but not guaranteed) if the method is dual

consistent. Certain objective functions and boundary conditions can lead to the continuous adjoint problem not being well posed [32, 33]. Load and displacement transfer schemes in aeroelasticity are often formulated discretely from nodes and/or elements and may create ill-posed problems for a continuous adjoint formulation. This remains unexplored in the literature. Whereas, the discrete adjoint approach still works because it only involves differentiating the existing primal problem code.

A final method of calculating sensitivity information is automatic or algorithmic differentiation (AD) [34, 35]. The user writes the code that implements forward problem or residual. The derivative of this forward problem code is not hand-derived but created either by operator overloading in the forward evaluation code or with source code transformation where a tool generates new functions that calculates derivatives of the functions in the original code. The chain rule is applied to find the total derivative from the individual functions of the full analysis. The chain rule accumulation can be in the same or reverse order compared to the forward analysis. These options have similar properties in terms of scaling with design variables and functions as the tangent and adjoint methods, respectively. Full automatic differentiation of a solver is typically less computationally efficient than a hand-differentiated tangent or adjoint method, but hand-differentiated adjoint methods requires a substantial effort to derive and implement in CFD solvers. Unlike the Jacobian in an implicit CFD solvers where approximations are suitable, the Jacobians in Equations 1.7 and 1.10 need to be exact to get accurate sensitivities. To reduce the effort required to implement an adjoint or tangent method, AD can be combined with the discrete adjoint or discrete tangent methods, i.e., using AD to calculate some or all of the parts of the partial derivatives in Equations 1.7-1.11. Muller and Cusdin [36] demonstrated that the computational cost of this method can be competitive to hand-coded discrete adjoints if the residual is implemented following specific rules.

1.4 Aeroelastic Design Optimization with Lower Fidelity Aerodynamics

Linear aeroelastic optimization tools have been computationally viable for decades [37–42]. Many of these tools are based on classical flutter analysis and linear aerodynamic methods such as vortex lattice [43] or doublet lattice [44] potential flow models. Their low computational cost has permitted researchers to explore sophisticated aeroelastic optimizations that include aeroelastic tailoring of composite wings [45], aeroservoelastic optimization [46], topology optimization [47], parametric uncertainty [48], and continuous gust models [49, 50]. Compared to the standard linear aerodynamic models, transonic small disturbance methods offer some improvement in the aerodynamic modeling of transonic flows because they retain some nonlinearity in the aerodynamics. Kolonay and Yang [51] demonstrated that optimizations with linear aerodynamics in transonic flow were nonconservative and differed considerably compared to a transonic small disturbance model. This is a very important consideration since many commercial aircraft operate in the transonic regime. However even transonic small disturbance codes have assumptions like reducing a wing cross-section to a camber line that affect the accuracy of the results. This motivates the use of Euler-based and Navier-Stokes-based CFD for optimizations where the other tools are not sufficient.

1.5 CFD-based Methods for Aeroelastic Optimization

Because three-dimensional CFD simulations require significant computational resources and time on the order of hours or days, it is desirable to reduce the number of simulations required to perform a design optimization. Therefore, CFD-based optimizations typically rely on either gradient-based methods (specifically adjoint-based sensitivities) to accommodate many design variables or some form of a reduced order or surrogate model to approximately represent the high-fidelity model.

1.5.1 Reduced Order Models and Surrogate Modeling

Reduced Order Models (ROMs) compress the CFD problem, which may have millions of states, into a model with tens or hundreds of degrees of freedom. These methods can construct the ROM based on system identification and Volterra theory from results from full CFD runs [52] or from linearized flow solutions about a nonlinear mean such as the Proper Orthogonal Decomposition (POD) technique [53, 54].

While the ROMs are significantly faster than full CFD solutions, they rely on superposition and/or linearization of the flow field about the equilibrium. When that nonlinear equilibrium changes, e.g., movement of shocks on the wing due a shape change during optimization, the linear assumption is not valid. If design changes significantly affect the nonlinear equilibrium or the aeroelastic characteristics are very sensitive to the equilibrium point, the ROM will be an inaccurate extrapolation or need to be reformed, meaning more full CFD runs and reduced advantage over standard CFD analysis.

Surrogate modeling is an alternative method to generate a less expensive model based on CFD. Rather than trying to reduced the model space of the problem like ROMs, surrogate modeling treats the governing equations as unknown and focuses on techniques that are generalizations of curve fitting and directly relate a set of inputs to a set of outputs. Common techniques include response surfaces generation [55], kriging [56, 57] and artificial neural networks [58]. Regardless of the method, the accuracy of the surrogate model is dependent on the number of full simulations performed as well as their location design space. For larger design spaces or scenarios where the flow conditions are very sensitive to design changes, a large number of necessary full simulations may eliminate the cost advantages of using a surrogate model but still allow a more thorough exploration of design space compared to gradient-based methods.

1.5.2 Steady Aeroelastic Adjoint Methods

In the past two decades, there have been numerous examples of steady CFD combined with finite element structures for aeroelastic optimization [59–69]. One of the first attempts at using CFD for aeroelastic optimizations was Maute et al. [60] who presented an adjoint and tangent method approach for three field formulation (fluid mesh coordinates, fluid states, and structural states). They coupled unstructured Euler CFD with finite element models that included composite plates and more detailed ones that include spars and ribs. For the composite plates, the optimal fiber orientations to maximize the lift-to-drag ratio were found for wings with Mach numbers between 0.4 and 1.2. For the wing with the more detailed finite element model, the lift-to-drag ratio was maximized subject to constraints on the lift, tip deflection, and von Mises stress in the stiffeners. The wing lift-to-drag ratio was improved by 19.44% by increasing the sweep of the wing to reduce wave drag and adjusting the twist of the wing.

Martins et al. [61] coupled structured Euler CFD with a linear shell element structure to optimize a supersonic business jet. They introduced aggregate constraints to reduce the number of required adjoint solutions. Martins et al. showed that coupled aero-structural optimization reduced the structural weight of the aircraft by an additional 16% compared to sequential optimization where aerodynamic optimization and structural optimization are performed iteratively but separately. The steady aeroelastic analysis of Martins et al. has since been extended to include Reynolds-averaged Navier-Stokes (RANS) CFD and more complex design cases with multipoint optimization [67].

These tools have proven effective for optimization of a range of vehicles with a large number of design variables even in complex conditions such as transonic flow. These studies typically produce slender, high-aspect-ratio designs which renders them more susceptible to adverse aeroelastic effects; however, these studies do not consider dynamic aeroelasticity in their analysis or design formulation, and lower-fidelity optimizations [50] have demonstrated that inclusion of flutter and gust constraints would likely reduce the gains ob-

served in these steady aeroelastic optimizations. For example, Kenway et al. [67] presented a multi-point optimization of a full transport aircraft with 476 design variables controlling the aerodynamics, structure parameters, and planform and airfoil shape. A fuel-burn optimization increased the wing aspect ratio from 9.0 to 12.6 while only increasing the weight of the wing by 2.3%. Kenway et al. simulated the response of their optimized design to a gust, and multiple structural components were above the failure condition. This indicates that while the steady coupled adjoint method is an effective optimization tool, unsteady aeroelastic effects and loading conditions must be included in constraints to produce more feasible designs.

One of the first applications of CFD to adjoint-based flutter suppression was by Palaniappan et al. [70] and was based on a steady adjoint method. Rather than directly apply an unsteady aeroelastic adjoint, Palaniappan et al. formed a linear representation of the aerodynamics with a least-squares fit of unsteady Euler simulations. A steady aerodynamic adjoint method approximated the unsteady linear aerodynamic model's lift and moment sensitivity with respect to blowing and suction at various locations on an airfoil. This method was applied to development of a feedback controller for flutter suppression of a pitching and plunging airfoil by means of blowing and suction. The optimal feedback controller generated via adjoint-based control design with the objective function being the minimization of the displacement and velocity at the final time of the simulation. The feedback controller was then tested in the full unsteady Euler analysis, and it was shown that the blowing and suction successfully suppressed flutter for an airfoil initially near the flutter point.

1.5.3 Time-accurate Adjoint Methods

Because many flow fields of interest are unsteady, there are multiple examples of researchers developing and applying time-accurate adjoint-enabled CFD solvers [71–75]. The time-accurate adjoint method is very general, but the chaotic systems cause the stan-

dard adjoint method to diverge [76]. Wang [77] extended the adjoint method to chaotic systems using a technique known as least-squares shadowing. While this method works, it is extremely computationally intensive even for relatively small problems. There is ongoing research to reduce the computational cost of this method [78].

In recent years, some research groups have extended time-accurate adjoint methods to aeroelastic problems. The unstructured CFD code NSU3D has been applied in various efforts to unsteady optimization. In Mani and Mavriplis [79], adjoint sensitivities were determined for two-dimensional airfoil problems with pitch and plunge springs. This unsteady coupled adjoint was then applied to aerodynamic shape optimizations where the goal was to suppress an initially divergent flutter of the airfoil by minimizing the square of the displacement at the final time step. A second optimization was performed minimizing the square of the displacements over the final ten steps (with approximately 36 times per oscillation period). From this work, Mishra et al. [80] extended the coupling with NSU3D to perform optimization of a flexible rotor in hover. The structural model was a bending and twisting beam coupled to the RANS equations in NSU3D. The design variables were shape parameterization variables, but these shape variables only affected the outer mold line of the blade seen by the aerodynamics with no effect on the structural model. The rotor was simulated for one revolution and the optimization minimized the rotor torque subject to a constrained thrust. Fabiano [81] and Fabiano and Mavriplis [82, 83] extended this work further to perform aeroacoustic optimization of flexible rotor blades in forward flight. From the aeroelastic solution, a Ffowcs Williams-Hawkings acoustic tool [84] was used to propagate sound pressure levels to the farfield. They presented a series of optimizations for the HART-II rotor. The design variables were the control inputs and the shape variables, but again these shape variables had no effect on the structural model. The objectives and constraints were related to acoustic noise and rotor torque, thrust, and moments. In one such optimization [83], Fabiano and Mavriplis demonstrated a torque reduction of 2.5% while decreasing the overall sound pressure level by 2 dB.

Wang et al. [85] performed aeroelastic sensitivity analysis by coupling Dymore [86] and FUN3D [87]. They combined complex sensitivities of beam models in Dymore with FUN3D's discrete adjoint to calculate sensitivities of aerodynamic quantities with respect to aerodynamic and shape variables. As in Mishra et al., these shape variables only affected the aerodynamic properties of the blades and not the structural model. Work is currently underway to replace the complex sensitivities in Dymore with an adjoint implementation to make the sensitivity analysis more efficient [88].

Zhang et al. [89–91] developed a time-accurate coupled adjoint capability for the Euler solver, ZEUS. The Cartesian meshes in ZEUS are stationary and transpiration boundary conditions account for the wing thickness and the motion of the wing. This limits the method to small amplitude motion. Zhang et al. [89] applied ZEUS to a similar optimization as Mani and Mavriplis where flutter of a pitch and plunge airfoil was suppressed by minimizing the lift coefficient at the end of the time-accurate simulation. The authors also developed a damping constraint where the damping is estimated with envelope functions (envelope functions will be described in detail in Chapter 4). They then demonstrated that this constraint can be applied to gradient-based design. In the optimization, an initially fluttering airfoil was transformed into one at its flutter point (neutrally stable oscillations) by changing its shape. In References [90] and [91], Zhang et al. extended this method to three-dimensional analysis with modal structural models and include structural design variables and the effect of shape variables through a combination of adjoint and complex-step-based sensitivities. The updated method was applied to mass minimization of the Goland wing model and a supersonic tailless configuration.

1.5.4 Frequency Domain Methods

Because high-fidelity time-accurate simulations have a significant computational cost, some research groups are pursuing frequency domain methods as an alternative. Although the scope of applicability is more limited than the time-accurate approach, CFD-based fre-

quency methods can be more efficient for problems that are periodic in nature. In linearized frequency domain methods [92], a nonlinear static aeroelastic solution is found. Then the problem is linearized about that point and converted into the Fourier domain. This method is efficient for linear stability problems such as flutter analysis. Widhalm et al. [92] demonstrated between 11 and 17 times reduction in computational time compared to time-domain simulations. Because it is a linearization, the method can lose accuracy in nonlinear problems when the motions are not small.

Others have pursued methods like the harmonic balance method [93–95] where the solution is assumed takes the form of a Fourier series without linearizing the governing equations. Like the linearized frequency domain methods, harmonic balance methods can be more computationally efficient than time-domain CFD. This method permits nonlinear unsteadiness so that cases such as transonic limit cycle oscillations [93, 94] can be accurately modeled. Once the periodic assumption is applied to the unsteady equations, the problem resembles a steady CFD system of equations where techniques such as pseudo-time marching can be used to solve the problem.

Many aeroelastic problems are periodic in nature which makes these frequency domain methods particularly useful. Although the frequency domain methods can be applied to problems that are not periodic, these problems may require multiple frequencies to accurately represent the time response, increasing the size of the frequency domain problems. For example, a pulse gust or 1-cosine gust have an infinite range of frequencies. The frequency domain methods can approximate the problem with the dominate frequencies, but it is more computationally efficient to simulate the problem in the time domain when there are a large number of relevant frequencies [96].

1.6 Thesis Objectives

Overall Thesis Objectives

It is evident from the literature that there are many important considerations in aeroelastic optimization: shape design, structural sizing, performance, stress constraints, steady aeroelastic effects, unsteady aeroelastic phenomena, etc. To date, the methods in the literature have made simplifying assumptions or ignored certain aspects of the problem while they focus on others. The literature on steady aeroelastic optimization includes structural considerations but ignores the unsteady aeroelasticity. Wang et al. [85] and Fabiano and Mavriplis [83] focus on the time-accurate aeroelastic coupling for rotorcraft while ignoring structural design aspects and unsteady aeroelastic constraints such as flutter or gust response. Zhang et al. [91] performed flutter-constrained design but assume small displacements to apply the transpiration boundary conditions, and they utilize a modal representation of the structure. To meet the CFD 2030 grand challenge of MDAO of a highly flexible aircraft configuration, all of the these considerations and more need to be present during the same optimization. More general methods are available in lower-fidelity optimization tools, but situations exist where these methods cannot accurately capture the complex nonlinear physics associated with the aeroelasticity of systems. With next generation aircraft designs becoming more flexible, lower-fidelity tools will likely be insufficient to accurately model and design these vehicles. The primary objective of the research here is therefore to develop a high-fidelity design methodology that is more general and allows unsteady and nonlinear effects (including large deflections), structural design variables and constraints, and important aeroelastic constraints such as flutter and gust response within the same model. The proposed methodology is based on the discrete adjoint method to efficiently compute sensitivity information.

Thesis Outline

The present dissertation is organized as follows:

1. Chapter 2 describes the computational tools utilized and developed to achieve the stated goals of the dissertation.

2. The aeroelastic coupling methodology is described in Chapter 3. The steady and unsteady forward analysis methodologies are described, and the corresponding adjoint formulations and sensitivity expressions are derived. After the coupling procedures are established, the coupling methodology and implementation are verified against some aeroelastic benchmark problems. The adjoint-based sensitivities are then verified against the complex-step finite-difference results.

3. Chapter 4 details the development of two constraint classes for time-domain aeroelastic optimization. The first constraint is a time-domain flutter constraint based on the matrix pencil method for estimation of the damping of the system. The second constraint is formulated for gust response based on the field velocity method of gust modeling.

4. Finally in Chapter 5, the coupling framework is demonstrated on a systematic series of aeroelastic design optimization problems. First, an energy harvesting optimization is performed on the vortex-induced vibration of a cylinder which is a complex interaction of dynamics of the cylinder and the vortex shedding in its wake. Next, a set of optimizations of the undeformed Common Research Model (uCRM) illustrate the application to aerospace design. The first uCRM optimization is performed with steady aeroelastic analysis and constraints to demonstrate that the coupling has equivalent capabilities to those in the high-fidelity static aeroelastic optimization literature. The second uCRM optimization is an adaption of the steady optimization where the wing root undergoes sinusoidal plunging. The final uCRM optimizations

are mass minimizations subjected to stress constraints from gust analyses. The last two sets of optimizations illustrate that the time-domain aeroelastic adjoint is applicable to aerospace configurations.

Broader Impact

The present work advances the state of the art by proposing a unified methodology for adjoint-based optimization of aeroelastic vehicles. It adds generality and structural model complexity compared to existing unsteady aeroelastic adjoint-based methods and extends the state of the art by proposing formulations for two aeroelastic phenomena: flutter and gust response. These aeroelastic constraints are critical for the design of next-generation aircraft, in which thin, slender, high aspect-ratio wings are being selected for performance considerations, making these designs more susceptible to adverse aeroelastic phenomena. With the high-fidelity tools described in this dissertation, the fidelity of design can be improved by considering aeroelasticity in complex situations including flutter in transonic flows, helicopter maneuver loads, and nonlinear response to gusts.

CHAPTER 2
COMPUTATIONAL TOOLS

2.1 FUN3D

The aerodynamics module to demonstrate in this work is the NASA Langley CFD code, FUN3D [87, 97]. FUN3D is a unstructured node-centered solver that handles both the compressible Navier-Stokes problems as well as incompressible flows with the method of artificial compressibility [98]. The flow field in FUN3D is modeled with the Arbitrary Lagrangian-Euler (ALE) formulation of the Navier–Stokes equations [99]. The residuals for the aerodynamic governing equations are obtained from a discrete approximation of the following integral over each finite volume:

$$\mathbf{A}\left(\mathbf{x},\mathbf{q},\dot{\mathbf{q}},\mathbf{x}_G,\dot{\mathbf{x}}_G\right)=\frac{\partial\left(\mathbf{q}V\right)}{\partial t}+\oint_{\partial V}\left(\overline{\overline{\mathbf{F}}}^{*}-\overline{\overline{\mathbf{F}}}_{\mathbf{v}}\right)\cdot\hat{\mathbf{n}}dS=0, \tag{2.1}$$

where $\mathbf{A}$ is the aerodynamic residual; $\mathbf{q}$ is the aerodynamic state vector; V is the volume of the dual cell; $\mathbf{x}_G$ are the aerodynamic volume mesh coordinates; $\overline{\overline{\mathbf{F}}}^{*}$ is the convective flux term; and $\overline{\overline{\mathbf{F}}}_{\mathbf{v}}$ is the diffusive flux term that includes viscosity and turbulence model effects. This equation is discretized in a node-based manner with the convective fluxes calculated with a least-squares-based flux-splitting scheme. The diffusive fluxes that include viscous and turbulence model effects are computed with a combination of an edge-based and Green-Gauss gradients. For unsteady simulations, an optimized second-order backwards difference formulation (BDF2opt) as well as other BDF methods are coupled with dual time-stepping to ensure converged solutions [99]. For turbulence closure, FUN3D includes the Reynolds-Averaged Navier-Stokes (RANS) equations with the one-equation Spalart-Allmaras (SA) [100] and the two-equation $k-\omega$ shear-stress transport (SST) model [101], as well as hybrid Reynolds-Averaged Navier-Stokes/Large Eddy Simulation

(HRLES) modeling, including the detached eddy simulation [102], delayed detached eddy simulation [103] and the hybrid RANS-LES model of Sanchez-Rocha et al. [104] with the Smagorinsky assumptions [105] in the sub-grid scales.

Moving grids in FUN3D can be treated with rigid motion, deforming motion, overset grids, or a combination of these methods. Rigid motion is dictated by a 4x4 rigid transform matrix that includes a 3x3 rotation matrix and a translation vector

$$\mathbf{T} = \begin{pmatrix} R_{11} & R_{12} & R_{13} & t_1 \\ R_{21} & R_{22} & R_{23} & t_2 \\ R_{31} & R_{32} & R_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2.2}$$

When a node's coordinates ($[x, y, z, 1]^T$) are multiplied by this transform matrix, the node is rotated about the origin according to the rotation matrix and then shifted by the translation vector. Rigid motion transforms can be prescribed, calculated by a six degree of freedom dynamics module, or determined by an external dynamics code such as Dymore [86] or CAMRAD [106].

When bodies of interest in the flow deform, the elastic motion of the surface needs to propagate to the volume mesh of the flow solver to prevent negative cell volumes and reduce poor quality elements. The mesh deformation in FUN3D is based on a linear elasticity model [26]. In this approach the volume mesh deformation is driven by the motion of the surface mesh. The displacements of the volume nodes are governed by

$$\nabla^2 \mathbf{V} + \frac{1}{1-2\nu} \nabla (\nabla \cdot \mathbf{V}) = 0. \tag{2.3}$$

This system of equations is discretized with a finite volume method to form the residual equation,

$$\mathbf{G}(x, \mathbf{x}_G, \mathbf{x}_A) \triangleq \mathbf{x}_{A0} + \mathbf{u}_A - \hat{\mathbf{x}}_A - \mathbf{K}_G (\mathbf{x}_G - \hat{\mathbf{x}}_G) = 0. \tag{2.4}$$

The terms with carets are the original mesh from the grid file; $\mathbf{x}_{A0}$ is the vector of deformed surface coordinates determined from the shape parameterization changes (fixed over a given simulation); $\mathbf{u}_A$ is the aeroelastic displacement vector for the aerodynamic surface; and $\mathbf{x_G}$ is the volume mesh(es) being solved for. The stiffness matrix, K_G, is computed from the original mesh from the grid file according to either the size of the cell or the distance from the surface. When the rigid motion transform is included, this system of equations becomes

$$\mathbf{G}\left(x,\mathbf{x}_G,\mathbf{x}_A,\mathbf{T}\right) \triangleq \mathbf{T}\left(\mathbf{x}_{A0}+\mathbf{u}_{A*}-\hat{\mathbf{x}}_A\right)-\mathbf{T}\mathbf{K}_G\mathbf{T}^{-1}\left(\mathbf{x}_G-\mathbf{T}\hat{\mathbf{x}}_G\right)=0, \tag{2.5}$$

where $\mathbf{u}_{A*}$ are now the aeroelastic displacements in the rigid transform's local reference frame. This system of equations is solved during each FUN3D iteration before the step in the flow solver occurs.

The FUN3D software suite includes a hand-coded discrete adjoint that has been applied in design optimization [107] and steady output-based spatial adaptation [108]. The adjoint implementation includes separation of mesh sensitivities [109], noninertial reference frames [110], and dynamic overset grids under forced motion [72, 73]. Recent work by Wang et al. [85] added multibody (comprehensive) rotorcraft analysis capability to the FUN3D adjoint solver using the complex step method for Dymore's contributions to the adjoint problem. For steady problems without coupling, the flow adjoint equation is

$$\left[\frac{\partial \mathbf{A}}{\partial \mathbf{q}}\right]^T \psi_A = -\left[\frac{\partial f}{\partial \mathbf{q}}\right]^T. \tag{2.6}$$

Rather than directly solving this equation, it is solved with a time-like marching form:

$$\left(\frac{V}{\Delta t}\mathbf{I}+\left[\frac{\partial \mathbf{A}}{\partial \mathbf{q}}\right]^T\right)\Delta\psi_A^n = -\frac{\partial f}{\partial \mathbf{q}}-\left[\frac{\partial \mathbf{A}}{\partial \mathbf{q}}\right]^T\psi_A^n, \tag{2.7}$$

where the updated adjoint vector is then $\psi_A^{n+1} = \psi_A^n + \Delta\psi_A^n$. This time-like approach creates a more robust solver. Once the solution to the flow adjoint equation is obtained, the grid adjoint equation is solved (if necessary) using a generalized minimal residual (GMRES) algorithm.

As part of this dissertation, a more general aeroelastic coupling interface was added to the FUN3D Python wrapper to perform analysis and optimization coupled with external structural models. The new interface methods allows a Python code to set transform matrices and surface node displacements into FUN3D as well as extract surface forces. Adjoint interface methods that correspond to these data exchanges were also added to the Python wrapper. As part of extracting surface forces, a method to calculate nodal forces on the surface mesh and the corresponding linearizations for the adjoint were included in FUN3D. The residual related to this operation is

$$\mathbf{F}\left(\mathbf{x}, \mathbf{x}_G, \mathbf{q}, \mathbf{f}_A\right) = \mathbf{f}_A - \boldsymbol{\phi}\left(\mathbf{x}, \mathbf{x}_G, \mathbf{q}\right) = 0, \tag{2.8}$$

where $\mathbf{f}_A$ is the nodal force vector and $\boldsymbol{\phi}$ is the action of the force integration which is dependent on the flow state and the aerodynamic mesh.

2.2 TACS

Structural modeling is performed with TACS [63, 111], a parallel structural analysis code. TACS is an adjoint-enabled finite element solver capable of linear and geometrically nonlinear analysis with multibody dynamics capability. TACS achieves parallel scalability through domain decomposition. The general structural governing equation in TACS in residual form is:

$$\mathbf{S}(\mathbf{x}, \mathbf{u}_S, \dot{\mathbf{u}}_S, \ddot{\mathbf{u}}_S, \mathbf{f}_S, t) \triangleq \mathbf{M}\ddot{\mathbf{u}}_S + \mathbf{C}\dot{\mathbf{u}}_S + \mathbf{K}\mathbf{u}_S - \mathbf{f}_S = 0. \tag{2.9}$$

For steady problems, this residual simplifies to

$$\mathbf{S}(\mathbf{x}, \mathbf{u}_S, \dot{\mathbf{u}}_S, \ddot{\mathbf{u}}_S, \mathbf{f}_S, t) \triangleq \mathbf{K}\mathbf{u}_S - \mathbf{f}_S = 0. \tag{2.10}$$

For time-dependent problems, TACS has multiple time marching schemes [112] including backward difference formulas, the Newmark method, Adams-Bashforth-Moulton method, and diagonally implicit Runge-Kutta. The discrete adjoint in TACS is capable of solving the adjoint problems associated with static and time-accurate finite element problems. The adjoint implementation in TACS has been applied extensively for design optimization [68, 69, 112, 113].

2.3 FUNtoFEM

The coupling of FUN3D and TACS is performed with FUNtoFEM, a Python-based framework developed for both high-fidelity aeroelastic analysis and adjoint-based aeroelastic optimization [114, 115]. The modularity of FUNtoFEM was designed with the goals of being able to couple any adjoint-enabled flow and structural models and to permit its extension to other disciplines such as thermal analysis. FUNtoFEM also includes several load and displacement transfer techniques for aeroelastic coupling. In this section, the primary load and displacement transfer scheme are described as well as the implementation aspects of the FUNtoFEM Python coupling framework.

2.3.1 Matching-based Extrapolation of Loads and Displacements

The transfer of loads and displacements is performed with the method called Matching-based Extrapolation of Loads and Displacements (MELD) [116]. This transfer methodology does not require any grid connectivity information which renders the algorithm easily scalable and compatible with general structural models. It also does not require rotations from the structural models eliminating the need for concentrated moments in the load trans-

fer to maintain consistency.

Displacement Transfer

For the displacement transfer, each aerodynamic surface node is connected to the set of its n nearest structural nodes. The movement of the aerodynamic node is extrapolated from the translation and rotation of the cloud of n nearest structural nodes using a rigid link connecting the aerodynamic node to a weighted center of the structural node point cloud. For a given aerodynamic node, k, this problem is formulated as a weighted least-squares problem:

$$\sigma^2 = \sum_{i=1}^{n} w_i ||\mathbf{x}_{S0,i} + \mathbf{u}_{S,i} - \left(\mathbf{R}_k \mathbf{x}_{S0,i} + \mathbf{t}_k\right)||^2. \tag{2.11}$$

The least-squares problem determines the best fit for the motion of the cloud of structural nodes by finding the rotation matrix, $\boldsymbol{R}_k$, and translation vector, $\boldsymbol{t}_k$, that minimize σ^2. The weights in the equation are based on the distance between the aerodynamic node and the structural node:

$$\hat{w}_i = e^{-\beta ||\boldsymbol{x}_{A0,k} - \boldsymbol{x}_{S0,i}||^2}, \tag{2.12}$$

and the weights are normalized according to

$$w_i = \frac{\hat{w}_i}{\sum_{j=1}^{n} \hat{w}_j}, \tag{2.13}$$

where β is a weighting factor, and $\mathbf{x}_{A,j}$ is the location of the aerodynamic node. The least-squares problem in Equation 2.11 is solved using a singular value decomposition [117]. Once the best-fit rotation and translation are found, the displacement of the aerodynamic node is computed as

$$\mathbf{x}_{A0,k} + \mathbf{u}_{A,k} = \mathbf{R}_k \mathbf{x}_{A0,k} + \mathbf{t}_k. \tag{2.14}$$

In residual form, the displacement transfer residual is then given by

$$\mathbf{D}(\mathbf{x}, \mathbf{u}_S, \mathbf{u}_A) = \mathbf{u}_A - \xi(\mathbf{x}, \mathbf{u}_S, \mathbf{x}_{A0}, \mathbf{x}_{S0}) = 0, \tag{2.15}$$

where ξ is function representing the action of displacement transfer in Equation 2.14.

Load Transfer

The load transfer is derived from the conservation of the virtual work on the aerodynamic and structural surfaces:

$$\delta W_A = \delta W_S. \tag{2.16}$$

The structural virtual work can written as the sum over the structural nodes of a consistent force vector times the virtual structural displacement vectors:

$$\delta W_S = \sum_i^S \boldsymbol{f}_{S,i} \delta \boldsymbol{u}_{S,i}. \tag{2.17}$$

The aerodynamic virtual work can be formulated so that it is also a function of the virtual structural displacements:

$$\delta W_A = \sum_k^A \frac{\partial W_A}{\partial \boldsymbol{u}_{A,k}} \delta \boldsymbol{u}_{A,k} = \sum_k^A \boldsymbol{f}_{A,k}^T \delta \boldsymbol{u}_{A,k} = \sum_i^S \boldsymbol{f}_{A,k}^T \frac{\partial \boldsymbol{u}_{A,k}}{\partial \boldsymbol{u}_{S,i}} \delta \boldsymbol{u}_{S,i}. \tag{2.18}$$

Equating these expressions and utilizing the fact that the virtual displacements are arbitrary and nonzero leads to the expression for the consistent force vector at each structural node,

$$\boldsymbol{f}_{S,k} = \sum_i^A \boldsymbol{f}_{A,i}^T \frac{\partial \boldsymbol{u}_{A,i}}{\partial \boldsymbol{u}_{s,k}}, \tag{2.19}$$

where the partial derivative in this expression is derived from the displacement transfer. In

residual form, this load transfer is

$$\mathbf{L}(\mathbf{x}, \mathbf{u}_S, \mathbf{f}_A, \mathbf{f}_S) = \mathbf{f}_S - \eta(\mathbf{x}, \mathbf{x}_{A0}, \mathbf{x}_{S0}, \mathbf{u}_S, \mathbf{f}_A) = 0, \tag{2.20}$$

where η is the right hand side of Equation 2.19. Because this force transfer method is derived from the conservation of virtual work, it produces consistent forces and conserves energy across the fluid-structure interface [118, 119].

Rigid Motion Extraction

Combining the rigid motion option with the deforming mesh movement in FUN3D provides several advantages. Having a best-fit rigid motion minimizes the amount of deformation necessary for the volume mesh. This reduces the cost of the mesh deformation and minimizes the deterioration of mesh quality due to deformation. Additionally when using overset meshes, the 'rigid+deform' motion option allows large relative motions of elastic bodies such as flexible helicopter blades spinning above a fuselage. For these reasons, it is beneficial to break the global reference frame displacements from Equation 2.15 into a best fit rigid motion of the body and local deformations. The displacement transfer in MELD is a least-squares problem to find a best fit rotational and translation of a subset of structural nodes (Equation 2.11); therefore, the kernel of the displacement transfer can be applied to the aerodynamic surface to find a best fit rigid motion of the entire aerodynamic surface. The residual of this action is written as

$$\mathbf{R}(\mathbf{x}, \mathbf{u}_A, \mathbf{T}) = \mathbf{T} - \rho(\mathbf{x}, \mathbf{x}_{A0}, \mathbf{u}_A) = 0, \tag{2.21}$$

where ρ is the action of applying the least-squares problem to find the best fit transform matrix. Once the transform matrix is found, the local elastic deformation, $\mathbf{u}_{A*}$ can be found with

$$\mathbf{E}(\mathbf{x}, \mathbf{u}_A, \mathbf{T}, \mathbf{u}_{A*}) = \mathbf{x}_{A0} + \mathbf{u}_A - \mathbf{T}(\mathbf{x}_{A0} + \mathbf{u}_{A*}) = 0. \tag{2.22}$$

This process is applied separately to each body in the problem. After solving for the local deformation, the rigid transform matrix and local deformations can be passed to FUN3D to update its volume mesh(es) with Equation 2.5.

2.3.2 Coupling Framework

The coupling framework in FUNtoFEM is object-oriented Python and is designed to simplify problem set up and automate much of the data organization related to aeroelastic design optimization. The framework abstracts the aeroelastic analysis and design into two concepts, a model and a driver. The model defines the problem and holds the designs and coupling data in the optimization problem. The driver is the implementation of the coupling algorithm and orchestrates the data transfer between the various components in the forward and adjoint coupling algorithms. The driver also distributes the problem definition from the model to the disciplinary solvers and populates the model with the results, the function values and sensitivities.

The FUNtoFEM model consists of body and scenario objects as illustrated in Figure 2.1. A model can have multiple scenario objects that hold the information associated with a particular point in a multipoint design. For example, a design problem for optimizing an aircraft may have multiple cruise conditions as well as maneuver, flutter, and/or gust conditions for sizing the structure. One scenario object would be created for each simulated condition. In the scenario objects, variables such as the flow conditions (Mach number, angle of attack, free stream dynamic pressure, etc.) are set for the design point. The functions of interest such as lift, drag, or structural failure required from that scenario are also defined in the scenario object. A model can also have multiple body objects, e.g., the blades and fuselage of a rotorcraft. The body objects hold design and coupling information such as structural thickness variables or structural displacements of the body, and they also have an instance of the shape parameterization and a transfer scheme for load and displacements associated with the body. Associating the transfer scheme with the body

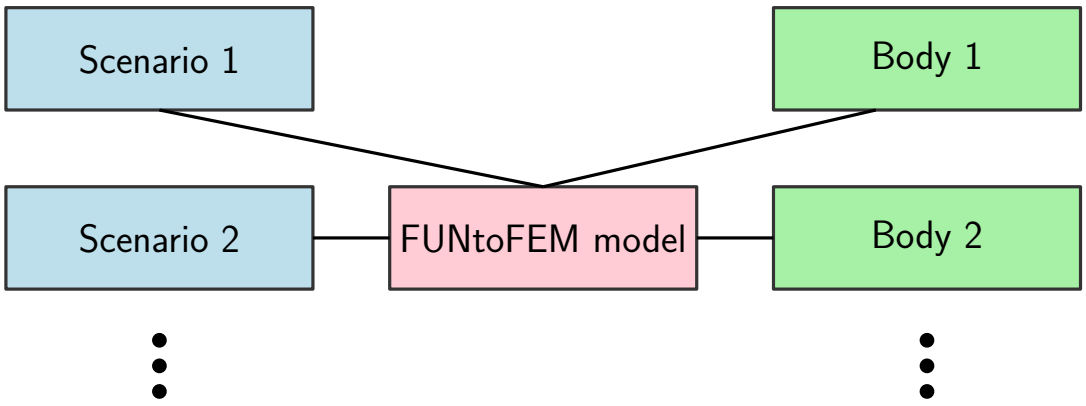


Figure 2.1: FUNtoFEM model composition.

object allows different bodies in the model to utilize different transfer schemes within the same simulation.

An example FUNtoFEM model for fixed wing design is given in Figure 2.2. In the example, the aircraft is assumed to be a half span model with a shell-based structure for the wing and a beam-based structure for the horizontal stabilizer. One body in the FUNtoFEM model is the wing which has a MELD transfer scheme instance and MASSOUD shape parameterization attached to it. The body object also has a dictionary of design variables that contains structural thickness and shape variables. The second body object for the horizontal stabilizer also has an associated MASSOUD parameterization, but since it is a beam model, it has a beam transfer scheme instead of MELD. Like the wing body, the horizontal stabilizer object contains a variable dictionary with structural and shape variables. The objective in this example is to minimize the drag at cruise subject to trim and stress constraints. For cruise performance, a scenario is defined with functions of interest being the drag for the objective, and lift and moment for trim constraints. For the cruise scenario, the optimizer can adjust the angle of attack. The stress constraints are set by two other scenarios, a gust and maneuver, which both have functions for structural failure. For the

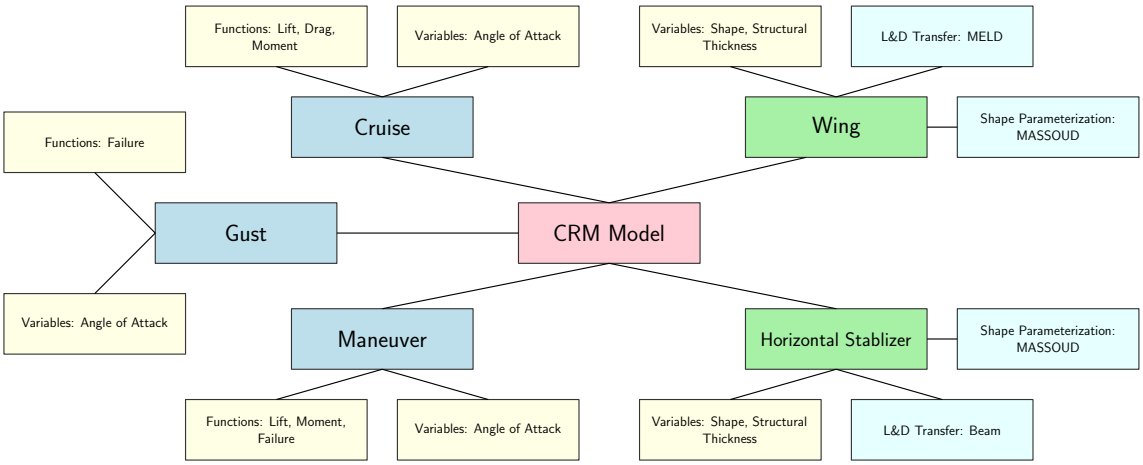


Figure 2.2: Example FUNtoFEM model for a fixed wing design.

gust condition, there is an angle of attack variable which can be independent or tied to the cruise angle of attack. Like the cruise condition, the maneuver includes lift and moment functions to trim the aircraft. Once this model defined, the driver will parse the model and pass the data to the disciplinary solvers. The driver can then loop over the scenarios to evaluate the functions of interest and their sensitivities. It will then populate the model with the function values and derivatives.

To reduce the computational time of the coupled solver, all data transfer in FUNtoFEM is performed in-core as opposed to file input/output. Parallelization with TACS, FUN3D, and the transfer scheme in FUNtoFEM is achieved with domain decomposition and MPI. In Figure 2.3, separate aerodynamic and structural MPI communicators are created. The FUNtoFEM driver utilizes the separate communicators to interact with the disciplinary solvers as indicated by the blue and orange lines in Figure 2.3. Because the transfer scheme requires both structural and aerodynamic surface data, the transfer scheme operates on the union the aerodynamic and structural communicators but still uses the individual communicators to access data from the disciplinary solvers.

FUNtoFEM defines a solver abstract base class to interact with disciplinary solvers. Solvers are added to FUNtoFEM by implementing the methods defined in the abstract base class. In these interface classes, generic functions are defined for the given solver, such as

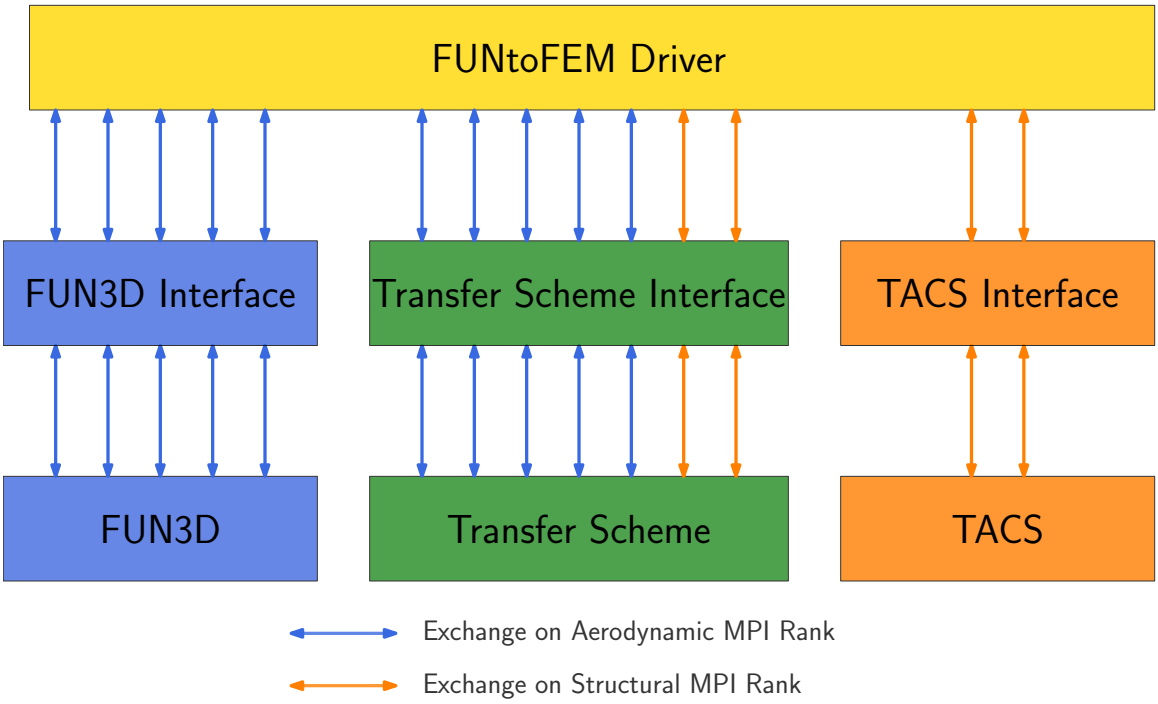


Figure 2.3: Parallel aeroelastic coupling in FUNtoFEM.

"initialize" or "iterate" functions. The level of abstraction is such that the framework does not know anything about the underlying solver other than it can call these generic functions and exchange data defined in the FUNtoFEM model. This leads to the ability to exchange a CFD solver with a vortex-lattice method [120], panel method [111], or even a different CFD solver. Likewise, a solid element FEM model can be exchanged with a shell or beam model, or a modal solver. This flexibility permits the assessment of how the quantities of interest change with solver fidelity for all parts of a multidisciplinary problem in addition to the mesh fidelity. New disciplines such as controls, thermal analysis, or acoustics can be added by creating an interface class for the new solver and implementing the new driver algorithm that couples the disciplines. To operate within FUNtoFEM, a Python wrapper for FUN3D (Fortran) is generated using f2py, and a Python wrapper is created for TACS (C++) with Cython. The solver interface classes in Figure 2.3 created for FUN3D and TACS then make calls to the solvers through these Python wrappers.

2.4 MASSOUD

Multidisciplinary Aerodynamic-Structural Shape Optimization Using Deformation (MASSOUD) [121] is a shape parameterization tool from NASA Langley Research Center. Rather than directly parameterizing the shape of wings, MASSOUD parameterizes the shape perturbations because almost all aerostructural optimizations begin with an existing wing design. This reduces the number of shape variables compared to a full parameterization of the geometry. Because the shape changes are small, the new disciplinary mesh(es) can be generated by mesh deformation rather than regenerating new grids for each new set of design parameters. Unlike many shape parameterization tools where the design variables are coordinates of control points, MASSOUD's design parameters are based on typical quantities used to define wing geometry. They include thickness, camber, twist angle, shear (dihedral) angle, as well as planform control point coordinates. Functions can be defined to relate the basic design variables, so that for example, planform control points can be related to the chord length, span, or sweep angle. MASSOUD uses soft object animation [122] and nonuniform rational B-spline (NURBS) representations to deform the geometry according a set of design variable values.

Within the FUNtoFEM framework, MASSOUD creates new aerodynamic surface meshes and structural meshes. The mesh deformation library within FUN3D updates the aerodynamic volume mesh(es). For the sensitivities of quantities with respect to the MASSOUD shape variables, MASSOUD provides the sensitivity of the node coordinates in the structural meshes and aerodynamic surface meshes to the shape variables, sometimes referred to as the design velocities. The sensitivities of the function with respect to the mesh coordinates and the design velocities are combined with the chain rule to compute the final sensitivities of the functions of interest with respect to the shape parameterization variables:

$$\frac{\partial f}{\partial x_{shape}} = \frac{\partial f}{\partial x_{A0}} \frac{\partial x_{A0}}{\partial x_{shape}} + \frac{\partial f}{\partial x_{S0}} \frac{\partial x_{S0}}{\partial x_{shape}}. \tag{2.23}$$

2.5 Hermes

The use of remote procedure calls allows a distributed application setup such as a client-server model, where the servers are different high performance computing clusters running different aspects of the multidisciplinary problem. There are two key aspects of the client-server model that make it attractive to include in a multidisciplinary design framework. The first is that the modularity enforced by the language independent interface definitions. As long as the interface definition is met, swapping servers is opaque to the client, so swapping CFD or structural solvers, or changing the fidelity of one component of the model and does not affect the other servers or require the coupling algorithm to change in the client. Another attractive feature of remote procedure calls is allowing servers that appear as black boxes to the user in addition to the server client. Organizations with proprietary models can set up servers that allow other groups to utilize the model by only exposing the interface. For example, an engine manufacturer may want keep details of an propulsion system model confidential but permit a fixed wing design team to utilize that model via a server.

Hermes [123] is cross-language remote procedure call framework built on the ZeroMQ library [124]. Given an interface definition, the Hermes compiler is designed to generate a server template in C++, Fortran, or Python. The developers then complete the server implementation by filling in the server template with calls to their analysis tool. The Hermes compiler then creates an executable for the server. Once this server program is started, a calling program can make request to the server over a network.

Hermes has been applied in FUNtoFEM to perform aeroelastic optimizations with the model shown in Figure 2.4. Servers are generated for the aerodynamic solver, structural solver, and transfer scheme. As opposed to Figure 2.3, the solver and transfer scheme interface act as clients instead of directly calling Python wrappers of the solvers and transfer scheme implementation. Serialization of the data for network communication represents a serious bottleneck because of the large amount of data that must be transferred in the high-

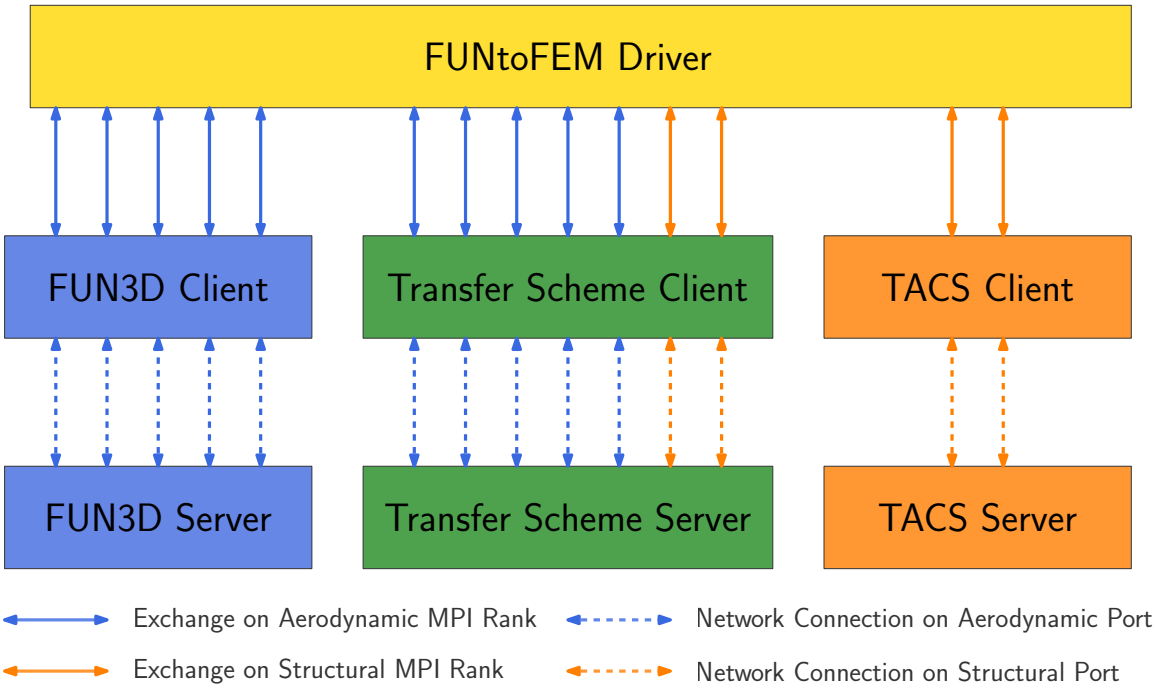


Figure 2.4: Hermes client-server model in FUNtoFEM.

fidelity problem. Therefore, the client server model has been implemented to maintain the distributed representation of the aerodynamic and structural vectors. Each server is started as an MPI process where every rank listens on a separate port, i.e., rank i of the server will listen on port $base\ port + i$. The FUNtoFEM driver is then started as an MPI process with a global communicator that is the same size as the transfer scheme server, an aerodynamic communicator that is the same size as the aerodynamic server, and a structural communicator that is the same size as the structural server. The FUNtoFEM driver exchanges data with and gives instructions to the clients. Each MPI rank of the clients then sends a request to the corresponding port of the server as represented by the dashed lines in Figure 2.4.

CHAPTER 3
PROPOSED AEROELASTIC COUPLING STRATEGY

The chapter describes the methodologies for steady and time-accurate aeroelastic problems developed to evaluate functions of interest and calculate their sensitivities with respect to design variables. After the coupling methodologies are outlined and the adjoint-based sensitivity expressions are derived, the forward analysis is verified with some standard aeroelastic test cases. Finally, the adjoint-based sensitivities are verified by comparing to the complex step method.

3.1 Steady Aeroelastic Adjoints

3.1.1 Forward Coupling Methodology

The aeroelastic coupling procedure is outlined in Algorithm 3 and the flow chart in Figure 3.1. For simplicity, the dependence on design variables is implied and has been omitted from the residual equations in Algorithm 3. The outlined algorithm represents a nonlinear block Gauss-Seidel method where each discipline is solved sequentially utilizing the most up-to-date data for coupling inputs. The iterative procedure starts by transferring the initial structural displacements to the aerodynamic surface with Equation 2.15. The volume mesh in the CFD problem is deformed to conform to the new surface (Equation 2.4). An iteration of the aerodynamic governing equations in Equation 2.1 is performed based on the new volume mesh. With Equation 2.8, new aerodynamic loads on the aerodynamic surface are calculated based on the new states and deformed mesh. The aerodynamic loads are then transferred to the structure with Equation 2.20. The structural problem, Equation 2.10, is solved resulting in deflection of the structure. These new structural deflections are passed to the displacement transfer and the process is repeated until the solution converges. Aitken

acceleration [125] is applied to the structural displacements; this under-relaxation provides stability for the nonlinear block Gauss-Seidel algorithm. For the steady analyses, the initial value of the under-relaxation factor, θ, was 0.125 while the minimum allowable value of the factor, θ_{min}, was 0.01.

The coupling algorithm could be modified into a block Jacobi method by only using the coupling input information at step $k-1$ to solve the residual equations at step k rather than utilizing the information at k when available. Lagging all of the states would allow all of the residuals to be evaluated simultaneously rather than sequentially making the coupled solver run faster per time step, but it also could affect the solver's stability and convergence rate.

Algorithm 3 Nonlinear block Gauss-Seidel method for steady aeroelastic analysis

Given: k_{max}, θ, θ_{min}

$\mathbf{u}_S^{(0)} \leftarrow 0$, $\Delta\mathbf{u}_S^{(0)} \leftarrow 0$ ▷ Set initial displacement vector and update vector to zero

for $k = 1$ to k_{max} **do**

$\mathbf{u}_A^{(k)} \leftarrow \mathbf{D}\left(\mathbf{u}_S^{(k-1)}, \mathbf{u}_A^{(k)}\right) = 0$ ▷ Transfer displacements to the aerodynamic surface

$\mathbf{x}_G^{(k)} \leftarrow \mathbf{G}\left(\mathbf{x}_{A0}, \mathbf{u}_A^{(k)}, \mathbf{x}_G^{(k)}\right) = 0$ ▷ Deform the volume mesh

$\mathbf{q}^{(k)} \leftarrow \mathbf{A}\left(\mathbf{q}^{(k)}, \mathbf{x}_G^{(k)}\right) = 0$ ▷ Evaluate the aerodynamics

$\mathbf{f}_A^{(k)} \leftarrow \mathbf{F}\left(\mathbf{x}_G^{(k)}, \mathbf{q}^{(k)}, \mathbf{f}_A^{(k)}\right) = 0$ ▷ Integrate loads over aerodynamic surface

$\mathbf{f}_S^{(k)} \leftarrow \mathbf{L}\left(\mathbf{u}_S^{(k-1)}, \mathbf{f}_A^{(k)}, \mathbf{f}_S^{(k)}\right) = 0$ ▷ Transfer the loads to the structure

$\mathbf{u}_S^{*} \leftarrow \mathbf{S}\left(\mathbf{u}_S^{*}, \mathbf{f}_S^{(k)}\right) = 0$ ▷ Evaluate the structures

$\Delta\mathbf{u}_S^{(k)} = \mathbf{u}_S^{*} - \mathbf{u}_S^{(k-1)}$ ▷ Calculate change in structural displacements

$\theta = \theta\left(1 - \frac{\left(\Delta\mathbf{u}_S^{(k)} - \Delta\mathbf{u}_S^{(k-1)}\right)\cdot\Delta\mathbf{u}_S^{(k)}}{\left\|\Delta\mathbf{u}_S^{(k)} - \Delta\mathbf{u}_S^{(k-1)}\right\|^2}\right)$ ▷ Update the Aitken under-relaxation factor

$\theta = \max(\min(\theta, 1), \theta_{min})$ ▷ Bound the under-relaxation factor

$\mathbf{u}_S^{(k)} = \mathbf{u}_S^{(k-1)} + \theta\Delta\mathbf{u}_S^{(k)}$ ▷ Under-relax the displacements

end for

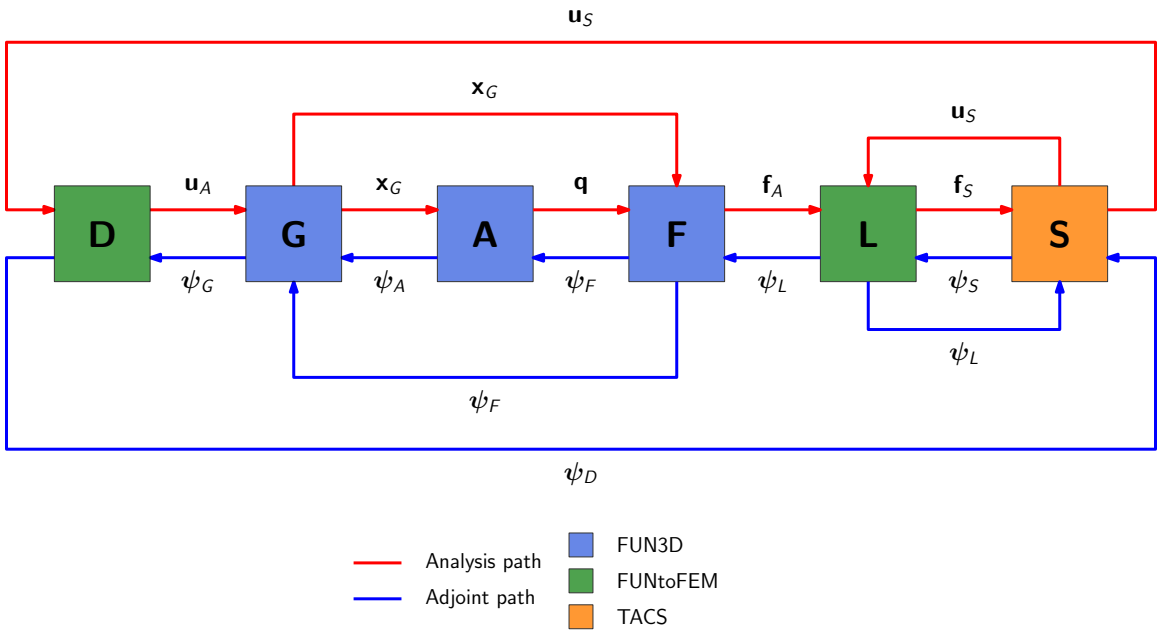


Figure 3.1: Illustration of the flow of information in the steady aeroelastic forward and adjoint analysis.

3.1.2 Coupled Adjoint Formulation

The flow of information in the adjoint problem is in the reverse direction of the forward analysis as illustrated by the blue lines in Figure 3.1. The coupled problem can be written as a single residual equation and adjoint vector as the adjoint method in Section 1.3 was described. However, splitting the residual and the adjoint vector into parts that represent different aspects of the forward problem has some important benefits. Primarily it permits the problem to be solved in a modular way analogous to the forward analysis described in Section 3.1.1. Additionally, by defining residuals for the coupling data transfers, the disciplinary solvers are not directly dependent on each other's states; they are only dependent on the coupling states. Since transfer schemes can be selected based on the type of structural model, removing the direct dependence of the aerodynamic residual on the structural states means that this present formulation is valid for virtually any type of structural discretization with no further modification required to the flow solver or its adjoint. Splitting the residual into disciplines also identifies a framework where straight forward implementation of

additional analyses such as acoustics, controls, or thermal considerations is possible with much of the mathematics and implementations remaining unchanged. With these goals in mind, the residuals were established so that each one has a single output state vector, i.e., the aerodynamic residual is only responsible for calculating the aerodynamic state, not the aerodynamic forces.

While the coupled adjoint equations could be derived in the same manner as in Section 1.3, here they are derived from a Lagrangian which results in the same equations but simplifies the derivation when there are multiple residuals. The Lagrangian contains the function of interest and the set of residuals from the forward coupling:

$$\begin{aligned}\mathscr{L}(\mathbf{x},\mathbf{q},\mathbf{x}_G,\mathbf{u}_A,\mathbf{f}_A,\mathbf{f}_S,\mathbf{u}_S) =& f(\mathbf{x},\mathbf{q},\mathbf{x}_G,\mathbf{u}_A,\mathbf{f}_A,\mathbf{f}_S,\mathbf{u}_S) + \psi_G^T\mathbf{G}(\mathbf{x},\mathbf{x}_G,\mathbf{u}_A)\\ &+ \psi_A^T\mathbf{A}(\mathbf{x},\mathbf{q},\mathbf{x}_G) + \psi_F^T\mathbf{F}(\mathbf{x},\mathbf{q},\mathbf{x}_G,\mathbf{f}_A)\\ &+ \psi_L^T\mathbf{L}(\mathbf{x},\mathbf{f}_A,\mathbf{f}_S,\mathbf{u}_S) + \psi_S^T\mathbf{S}(\mathbf{x},\mathbf{f}_S,\mathbf{u}_S)\\ &+ \psi_D^T\mathbf{D}(\mathbf{x},\mathbf{u}_A,\mathbf{u}_S).\end{aligned} \tag{3.1}$$

The adjoint equations can be formed as the Karush-Kuhn-Tucker (KKT) conditions [126, 127] of this Lagrangian, essentially setting the partial derivative of the Lagrangian with respect to each state variable equal to zero:

$$\mathbf{A}_\psi(\psi_A,\psi_F) = \left[\frac{\partial\mathbf{A}}{\partial\mathbf{q}}\right]^T\psi_A + \left[\frac{\partial\mathbf{F}}{\partial\mathbf{q}}\right]^T\psi_F + \left[\frac{\partial\mathbf{f}}{\partial\mathbf{q}}\right]^T = 0, \tag{3.2}$$

$$\mathbf{G}_\psi(\psi_A,\psi_G,\psi_F) = \left[\frac{\partial\mathbf{A}}{\partial\mathbf{x}_G}\right]^T\psi_A + \left[\frac{\partial\mathbf{G}}{\partial\mathbf{x}_G}\right]^T\psi_G + \left[\frac{\partial\mathbf{F}}{\partial\mathbf{x}_G}\right]^T\psi_F + \left[\frac{\partial\mathbf{f}}{\partial\mathbf{x}_G}\right]^T = 0, \tag{3.3}$$

$$\mathbf{D}_\psi(\psi_G,\psi_D) = \left[\frac{\partial\mathbf{G}}{\partial\mathbf{u}_A}\right]^T\psi_G + \left[\frac{\partial\mathbf{D}}{\partial\mathbf{u}_A}\right]^T\psi_D + \left[\frac{\partial\mathbf{f}}{\partial\mathbf{u}_A}\right]^T = 0, \tag{3.4}$$

$$\mathbf{F}_\psi(\psi_G,\psi_D) = \left[\frac{\partial\mathbf{F}}{\partial\mathbf{f}_A}\right]^T\psi_F + \left[\frac{\partial\mathbf{L}}{\partial\mathbf{f}_A}\right]^T\psi_L + \left[\frac{\partial\mathbf{f}}{\partial\mathbf{f}_A}\right]^T = 0, \tag{3.5}$$

$$\mathbf{L}_\psi(\psi_S,\psi_L) = \left[\frac{\partial\mathbf{S}}{\partial\mathbf{f}_S}\right]^T\psi_S + \left[\frac{\partial\mathbf{L}}{\partial\mathbf{f}_S}\right]^T\psi_L + \left[\frac{\partial\mathbf{f}}{\partial\mathbf{f}_S}\right]^T = 0, \tag{3.6}$$

$$\mathbf{S}_{\psi}(\psi_D, \psi_S, \psi_L) = \left[\frac{\partial \mathbf{D}}{\partial \mathbf{u}_S}\right]^T \psi_D + \left[\frac{\partial \mathbf{S}}{\partial \mathbf{u}_S}\right]^T \psi_S + \left[\frac{\partial \mathbf{L}}{\partial \mathbf{u}_S}\right]^T \psi_L + \left[\frac{\partial \mathbf{f}}{\partial \mathbf{u}_S}\right]^T = 0. \tag{3.7}$$

This general form of the adjoint equations can be simplified for the set of governing equations described in the forward analysis, Sect. 3.1.1:

$$L_{\psi}(\psi_L, \psi_S) = [\mathbf{I}]\,\psi_L - [\mathbf{I}]\,\psi_S = 0, \tag{3.8}$$

$$F_{\psi}(\psi_F, \psi_L) = [\mathbf{I}]\,\psi_F - \left[\frac{\partial L}{\partial \mathbf{f}_A}\right]^T \psi_L = 0, \tag{3.9}$$

$$A_{\psi}(\psi_A, \psi_F) = \left[\frac{\partial A}{\partial \mathbf{q}}\right]^T \psi_A + \left[\frac{\partial F}{\partial \mathbf{q}}\right]^T \psi_F + \left[\frac{\partial f}{\partial \mathbf{q}}\right]^T = 0, \tag{3.10}$$

$$G_{\psi}(\psi_A, \psi_G) = \left[\frac{\partial G}{\partial \mathbf{x}_G}\right]^T \psi_G + \left[\frac{\partial A}{\partial \mathbf{x}_G}\right]^T \psi_A + \left[\frac{\partial f}{\partial \mathbf{x}_G}\right]^T = 0, \tag{3.11}$$

$$D_{\psi}(\psi_G, \psi_D) = [\mathbf{I}]\,\psi_D + [\mathbf{I}]\,\psi_G + \left[\frac{\partial F}{\partial \mathbf{x}_A}\right]^T \psi_F = 0, \tag{3.12}$$

$$S_{\psi}(\psi_S, \psi_D, \psi_L) = \left[\frac{\partial S}{\partial \mathbf{u}}\right]^T \psi_S + \left[\frac{\partial D}{\partial \mathbf{u}}\right]^T \psi_D + \left[\frac{\partial L}{\partial \mathbf{u}}\right]^T \psi_L + \left[\frac{\partial f}{\partial \mathbf{u}}\right]^T = 0. \tag{3.13}$$

Initializing the structural adjoint variable, ψ_S, Equations 3.8-3.13 are iteratively solved until the adjoint solution converges using a linear block Gauss-Seidel method. The procedure is outlined in Algorithm 4. Similar to the forward analysis, Aitken under-relaxation is applied to the structural adjoint vector to provide stability. One advantage of this formulation and solution methodology for the adjoint problem is that each component can utilize the existing adjoint solver implementations for that discipline. FUN3D utilizes its time-like marching technique with improved robustness, and TACS solves the adjoint problem with an efficient generalized minimal residual (GMRES) solver. Only Jacobian-vector products that are the size of the coupling vectors in the forward problem (loads and displacements vectors) need to be passed between the existing solvers with these terms added to the right hand side of the existing adjoint equations.

Algorithm 4 Linear block Gauss-Seidel method for the adjoint solution

Given: k_{max}, θ, θ_{min}

$\psi_S^{(0)} \leftarrow 0, \Delta\psi_S^{(0)} \leftarrow 0$ ▷ Set initial structural adjoint vector and update vector to zero

for $k = 1$ to k_{max} **do**

$\psi_L^{(k)} \leftarrow \mathbf{L}_\psi\left(\psi_L^{(k)}, \psi_S^{(k-1)}\right) = 0$ ▷ Force transfer adjoint equation

$\psi_F^{(k)} \leftarrow \mathbf{F}_\psi\left(\psi_F^{(k)}, \psi_L^{(k)}\right) = 0$ ▷ Force integration adjoint equation

$\psi_A^{(k)} \leftarrow \mathbf{A}_\psi\left(\psi_A^{(k)}, \psi_F^{(k)}\right) = 0$ ▷ Aerodynamic adjoint equation

$\psi_G^{(k)} \leftarrow \mathbf{G}_\psi\left(\psi_G^{(k)}, \psi_A^{(k)}, \psi_F^{(k)}\right) = 0$ ▷ Grid deformation adjoint equation

$\psi_D^{(k)} \leftarrow \mathbf{D}_\psi\left(\psi_D^{(k)}, \psi_G^{(k)}\right) = 0$ ▷ Displacement transfer adjoint equation

$\psi_S^* \leftarrow \mathbf{S}_\psi\left(\psi_S^*, \psi_D^{(k)}, \psi_L^{(k)}\right) = 0$ ▷ Structural adjoint equation

$\Delta\psi_S^{(k)} = \psi_S^* - \psi_S^{(k-1)}$ ▷ Calculate change in structural adjoints

$\theta = \theta\left(1 - \frac{\left(\Delta\psi_S^{(k)} - \Delta\psi_S^{(k-1)}\right)\cdot\Delta\psi_S^{(k)}}{\left\|\Delta\psi_S^{(k)} - \Delta\psi_S^{(k-1)}\right\|^2}\right)$ ▷ Update the Aitken under-relaxation factor

$\theta = \max(\min(\theta, 1), \theta_{min})$ ▷ Bound the under-relaxation factor

$\psi_S^{(k)} = \psi_S^{(k-1)} + \theta\Delta\psi_S^{(k)}$ ▷ Under-relax the structural adjoint

end for

3.1.3 Adjoint-based Sensitivities

The expressions for the adjoint-based sensitivities are found by differentiating the Lagrangian in Equation 3.1 with respect to design variables. The general expression for the steady aeroelastic problem is given by

$$\frac{d\mathscr{L}}{d\mathbf{x}} \triangleq \nabla_{\mathbf{x}} f = \frac{\partial f}{\partial \mathbf{x}} + \psi_G^T \frac{\partial \mathbf{G}}{\partial \mathbf{x}} + \psi_A^T \frac{\partial \mathbf{A}}{\partial \mathbf{x}} + \psi_F^T \frac{\partial \mathbf{F}}{\partial \mathbf{x}} + \psi_L^T \frac{\partial \mathbf{L}}{\partial \mathbf{x}} + \psi_S^T \frac{\partial \mathbf{S}}{\partial \mathbf{x}} + \psi_D^T \frac{\partial \mathbf{D}}{\partial \mathbf{x}}. \tag{3.14}$$

Depending on the type of design variable, this expression can be simplified. The aerodynamic design variables, such as free-stream Mach number, angle of attack, or yaw angle are only directly influenced by the aerodynamic residual and potentially the function of

interest. Therefore, the sensitivity expression for aerodynamic variables is

$$\nabla_{\mathbf{x}} f = \frac{\partial f}{\partial \mathbf{x}} + \psi_A^T \frac{\partial \mathbf{A}}{\partial \mathbf{x}}. \tag{3.15}$$

This equation is valid whether the objective function is aerodynamic or related to some other state. Similarly, the dynamic pressure only directly affects the force integration residual leading to a dynamic pressure sensitivity expression of

$$\nabla_{\mathbf{x}} f = \frac{\partial f}{\partial \mathbf{x}} + \psi_F^T \frac{\partial \mathbf{F}}{\partial \mathbf{x}}, \tag{3.16}$$

and the structural variables only explicitly influence the structural residual:

$$\nabla_{\mathbf{x}} f = \frac{\partial f}{\partial \mathbf{x}} + \psi_S^T \frac{\partial \mathbf{S}}{\partial \mathbf{x}}. \tag{3.17}$$

This expression holds regardless of the type of structural design variable (Young's modulus, shell/panel thickness, beam cross-section properties, composite layup angle, etc.). Equations 3.15 and 3.17 are identical to the expressions from the uncoupled single discipline adjoint-based sensitivities. The difference is that the adjoint vectors in these equations are the result of the coupled adjoint problem and therefore carry the coupling sensitivity information into these derivatives.

Because they affect the aerodynamic surface and the initial structure, the sensitivities with respect to shape design variables requires terms from multiple residuals:

$$\nabla_{\mathbf{x}} f = \frac{\partial f}{\partial \mathbf{x}} + \psi_G^T \frac{\partial \mathbf{G}}{\partial \mathbf{x}} + \psi_L^T \frac{\partial \mathbf{L}}{\partial \mathbf{x}} + \psi_S^T \frac{\partial \mathbf{S}}{\partial \mathbf{x}} + \psi_D^T \frac{\partial \mathbf{D}}{\partial \mathbf{x}}. \tag{3.18}$$

Rather than direct differentiation of the residuals with respect shape variables, this expression is deconstructed with the chain rule into partial derivatives of the residuals with respect to aerodynamic surface and structural grid coordinates and partial derivatives of those grid

coordinates with respect to the shape design variables. The latter partial derivatives are sometimes referred to design velocities. Splitting the shape derivative with the chain rule avoids having any solver being tied directly to a particular shape parameterization. With this change applied to Equation 3.18, the sensitivities with respect to shape variables become

$$\begin{aligned}\nabla_{\mathbf{x}} f = &\left\{ \frac{\partial f}{\partial \mathbf{x}_{A0}} + \left[(\psi_D)^T \frac{\partial \mathbf{D}}{\partial \mathbf{x}_{A0}} + (\psi_L)^T \frac{\partial \mathbf{L}}{\partial \mathbf{x}_{A0}} + (\psi_G)^T \frac{\partial \mathbf{G}}{\partial \mathbf{x}_{A0}} \right] \right\} \frac{\partial \mathbf{x}_{A0}}{\partial \mathbf{x}} \\ &+ \left\{ \frac{\partial f}{\partial \mathbf{x}_{S0}} + \left[(\psi_D)^T \frac{\partial \mathbf{D}}{\partial \mathbf{x}_{S0}} + (\psi_L)^T \frac{\partial \mathbf{L}}{\partial \mathbf{x}_{S0}} + (\psi_S)^T \frac{\partial \mathbf{S}}{\partial \mathbf{x}_{S0}} \right] \right\} \frac{\partial \mathbf{x}_{S0}}{\partial \mathbf{x}}.\end{aligned} \tag{3.19}$$

Equation 3.19 is assembled by FUNtoFEM. The terms inside the curly brackets are calculated by the disciplinary solvers responsible for evaluating the associated residual or function, and the design velocities are provided by the shape parameterization tool.

3.2 Time-accurate Aeroelastic Adjoints

3.2.1 Time-accurate Coupling Methodology

The time-dependent forward analysis in Algorithm 5 extends the nonlinear block Gauss-Seidel algorithm from the steady analysis to the unsteady problem by lagging the structural displacements at each time step. The coupling is illustrated graphically in Figure 3.2. In this figure, each block represents an evaluation of a component residual and each row of blocks represents a time step. First, the initial conditions are set in the aerodynamic and structural solvers based on the design variables. The algorithm then loops over the specified number of time steps. Each time step begins by updating the aerodynamic surface displacements based on the structural displacements from the previous time step. The aerodynamic surface displacements are then split into a rigid transformation and local elastic deformations with Equations 2.21 and 2.22. Next, the aerodynamic volume mesh is rigidly moved and deformed to match the new surface according to Equation 2.5. The aerodynamic solution is then computed for the current time step. Pseudo-time subiterations are employed to ensure

Algorithm 5 Nonlinear block Gauss-Seidel method for the time-accurate analysis.

Given: k_{max}

$\mathbf{u}_S^{(0)} \leftarrow \mathbf{S}^{(0)}\left(\mathbf{u}_S^{(0)}\right)$ ▷ Set initial structural conditions

$\mathbf{x}_G^{(0)} \leftarrow \mathbf{G}^{(0)}\left(\mathbf{x}_G^{(0)}\right)$ ▷ Set the initial volume mesh(es)

$\mathbf{q}^{(0)} \leftarrow \mathbf{A}^{(0)}\left(\mathbf{q}^{(0)}, \mathbf{x}_G^{(0)}\right)$ ▷ Set the initial flow state

for $k = 1$ to k_{max} **do**

$\mathbf{u}_A^{(k)} \leftarrow \mathbf{D}\left(\mathbf{u}_S^{(k-1)}, \mathbf{u}_A^{(k)}\right) = 0$ ▷ Calculate the global aerodynamic displacements

$\mathbf{T}^{(k)} \leftarrow \mathbf{R}\left(\mathbf{u}_A^{(k)}, \mathbf{T}^{(k)}\right) = 0$ ▷ Extract the rigid transformation(s)

$\mathbf{u}_{A*}^{(k)} \leftarrow \mathbf{E}\left(\mathbf{u}_{A*}^{(k)}, \mathbf{u}_A^{(k)}, \mathbf{T}^{(k)}\right) = 0$ ▷ Extract the local deformation

$\mathbf{x}_G^{(k)} \leftarrow \mathbf{G}\left(\mathbf{u}_{A*}^{(k)}, \mathbf{T}^{(k)}, \mathbf{x}_G^{(k)}\right) = 0$ ▷ Deform the aerodynamic volume mesh(es)

$\mathbf{q}^{(k)} \leftarrow \mathbf{A}\left(\mathbf{q}^{(k)}, \dot{\mathbf{q}}^{(k)}, \mathbf{x}_G^{(k)}, \dot{\mathbf{x}}_G^{(k)}\right) = 0$ ▷ Evaluate the aerodynamics

$\mathbf{f}_A^{(k)} \leftarrow \mathbf{F}\left(\mathbf{x}_G^{(k)}, \mathbf{q}^{(k)}, \mathbf{f}_A^{(k)}\right) = 0$ ▷ Integrate loads over aerodynamic surface(s)

$\mathbf{f}_S^{(k)} \leftarrow \mathbf{L}\left(\mathbf{u}_S^{(k-1)}, \mathbf{f}_A^{(k)}, \mathbf{f}_S^{(k)}\right) = 0$ ▷ Transfer the loads to the structure

$\mathbf{u}_S^{(k)} \leftarrow \mathbf{S}\left(\mathbf{u}_S^{(k)}, \dot{\mathbf{u}}_S^{(k)}, \ddot{\mathbf{u}}_S^{(k)}, \mathbf{f}_S^{(k)}\right) = 0$ ▷ Evaluate the structures

end for

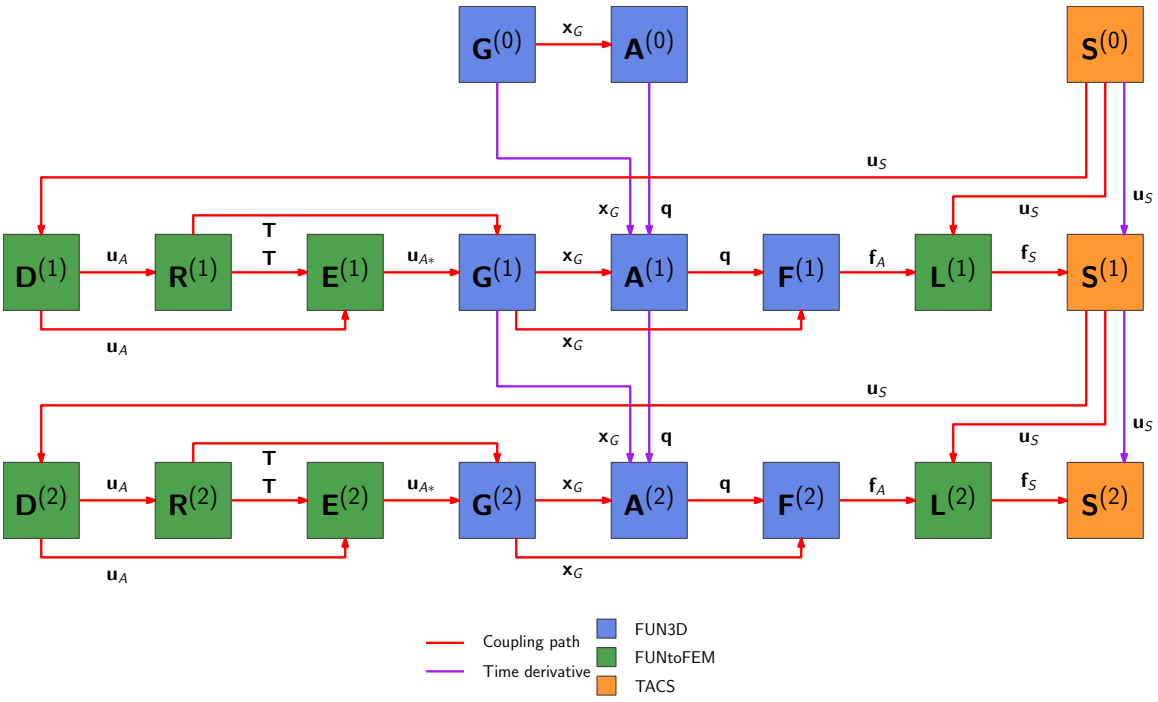


Figure 3.2: Flow of information during the time-accurate coupled forward analysis.

that the flow residual sufficiently converges at each time step. From the updated aerodynamic solution, forces are computed on the aerodynamic surfaces and then transferred to the structural model. The structural displacements are then updated with Equation 2.9 before moving onto the next time step.

The two residuals that contain time derivatives of state variables are the aerodynamic and structural residuals. In Figure 3.2, the derivatives in the aerodynamic residual are only functions of aerodynamic states and the aerodynamic volume mesh, and the time derivatives in the structural residual are only functions of previous structural states. In other words, the time derivative terms are contained within their respective solvers; therefore, for the current formulation, the aerodynamic and structural solvers can use any temporal discretization so long as the time step size is the same for both.

When lagging the structural displacement variable as proposed here, a simulation may exhibit numerical instability for very flexible structures due to apparent mass effects [128]. For the types of aerospace design applications for which this framework is being developed, this instability is not anticipated to be an issue as the mass ratio (density of the structure compared to the density of the fluid) is much greater than unity. However, if numerical instability does occur, the nonlinear block Gauss-Seidel algorithm can be modified to include fluid-structure interaction (FSI) subiterations which improve stability. FSI subiterations would create an additional solver loop over each row in Figure 3.2 to converge the coupled problem at each time step.

3.2.2 Time-accurate Adjoint Problem

The time-accurate adjoint is derived in the same manner as the steady aeroelastic problem. To derive the time-dependent coupled adjoint equations represented by the blocks in Figure 3.3, a Lagrangian is formed that includes the objective function, the residuals at every

time step, and the initial condition residuals:

$$\begin{aligned}\mathscr{L} = f + \sum_{k=1}^{N} \Bigg[& \left(\psi_D^{(k)}\right)^T \mathbf{D}^{(k)} + \left(\psi_R^{(k)}\right)^T \mathbf{R}^{(k)} + \left(\psi_E^{(k)}\right)^T \mathbf{E}^{(k)} + \left(\psi_G^{(k)}\right)^T \mathbf{G}^{(k)} \\ & + \left(\psi_A^{(k)}\right)^T \mathbf{A}^{(k)} + \left(\psi_F^{(k)}\right)^T \mathbf{F}^{(k)} + \left(\psi_L^{(k)}\right)^T \mathbf{L}^{(k)} + \left(\psi_S^{(k)}\right)^T \mathbf{S}^{(k)} \Bigg] \\ & + \left(\psi_G^{(0)}\right)^T \mathbf{G}^{(0)} + \left(\psi_A^{(0)}\right)^T \mathbf{A}^{(0)} + \left(\psi_S^{(0)}\right)^T \mathbf{S}^{(0)}.\end{aligned} \tag{3.20}$$

The adjoint equations are then based on the KKT conditions of the Lagrangian. At each time step k:

$$\begin{aligned}\mathbf{S}_\psi^{(k)} = & \left[\frac{\partial \mathbf{D}^{(k+1)}}{\partial \mathbf{u}_S^{(k)}}\right]^T \psi_D^{(k+1)} + \left[\frac{\partial \mathbf{L}^{(k+1)}}{\partial \mathbf{u}_S^{(k)}}\right]^T \psi_L^{(k+1)} + \left[\frac{\partial \mathbf{S}^{(k)}}{\partial \mathbf{u}_S^{(k)}}\right]^T \psi_S^{(k)} \\ & + \sum_{i=1}^{m_2} \left[\left(\frac{\partial \mathbf{S}^{(k+i)}}{\partial \dot{\mathbf{u}}_S^{(k+i)}}\right) \frac{\partial \dot{\mathbf{u}}_S^{(k+i)}}{\partial \mathbf{u}_S^{(k)}}\right]^T \psi_S^{(k+i)} \\ & + \sum_{i=1}^{m_3} \left[\left(\frac{\partial \mathbf{S}^{(k+i)}}{\partial \ddot{\mathbf{u}}_S^{(k+i)}}\right) \frac{\partial \ddot{\mathbf{u}}_S^{(k+i)}}{\partial \mathbf{u}_S^{(k)}}\right]^T \psi_S^{(k+i)} \\ & + \left[\frac{\partial f}{\partial \mathbf{u}_S^{(k)}}\right]^T = 0,\end{aligned} \tag{3.21}$$

$$\mathbf{L}_\psi^{(k)} = \left[\frac{\partial \mathbf{S}^{(k)}}{\partial \mathbf{f}_s^{(k)}}\right]^T \psi_S^{(k)} + \left[\frac{\partial \mathbf{L}^{(k)}}{\partial \mathbf{f}_s^{(k)}}\right]^T \psi_L^{(k)} + \left[\frac{\partial f}{\partial \mathbf{f}_s^{(k)}}\right]^T = 0, \tag{3.22}$$

$$\mathbf{F}_\psi^{(k)} = \left[\frac{\partial \mathbf{F}^{(k)}}{\partial \mathbf{f}_A^{(k)}}\right]^T \psi_F^{(k)} + \left[\frac{\partial \mathbf{L}^{(k)}}{\partial \mathbf{f}_A^{(k)}}\right]^T \psi_L^{(k)} + \left[\frac{\partial f}{\partial \mathbf{f}_A^{(k)}}\right]^T = 0, \tag{3.23}$$

$$\begin{aligned}\mathbf{A}_\psi^{(k)} = & \left[\frac{\partial \mathbf{A}^{(k)}}{\partial \mathbf{q}^{(k)}}\right]^T \psi_A^{(k)} + \sum_{i=1}^{m_1} \left[\left(\frac{\partial \mathbf{A}^{(k+i)}}{\partial \dot{\mathbf{q}}^{(k+i)}}\right) \frac{\partial \dot{\mathbf{q}}^{(k+i)}}{\partial \mathbf{q}^{(k)}}\right]^T \psi_A^{(k+i)} \\ & + \left[\frac{\partial \mathbf{F}^{(k)}}{\partial \mathbf{q}^{(k)}}\right]^T \psi_F^{(k)} + \left[\frac{\partial f}{\partial \mathbf{q}^{(k)}}\right]^T = 0,\end{aligned} \tag{3.24}$$

$$\mathbf{G}_{\psi}^{(k)} = \left[\frac{\partial \mathbf{G}^{(k)}}{\partial \mathbf{x}_G^{(k)}}\right]^T \psi_G^{(k)} + \left[\frac{\partial \mathbf{A}^{(k)}}{\partial \mathbf{x}_G^{(k)}}\right]^T \psi_A^{(k)} + \sum_{i=1}^{m_1} \left[\left(\frac{\partial \mathbf{A}^{(k+i)}}{\partial \dot{\mathbf{x}}_G^{(k+i)}}\right) \frac{\partial \dot{\mathbf{x}}_G^{(k+i)}}{\partial \mathbf{x}_G^{(k)}}\right]^T \psi_A^{(k+i)} + \left[\frac{\partial \mathbf{F}^{(k)}}{\partial \mathbf{x}_G^{(k)}}\right]^T \psi_F^{(k)} + \left[\frac{\partial f}{\partial \mathbf{x}_G^{(k)}}\right]^T = 0, \tag{3.25}$$

$$\mathbf{E}_{\psi}^{(k)} = \left[\frac{\partial \mathbf{E}^{(k)}}{\partial \mathbf{u}_{A*}^{(k)}}\right]^T \psi_E^{(k)} + \left[\frac{\partial \mathbf{G}^{(k)}}{\partial \mathbf{u}_{A*}^{(k)}}\right]^T \psi_G^{(k)} + \left[\frac{\partial f}{\partial \mathbf{u}_{A*}^{(k)}}\right]^T = 0, \tag{3.26}$$

$$\mathbf{R}_{\psi}^{(k)} = \left[\frac{\partial \mathbf{R}^{(k)}}{\partial \mathbf{T}^{(k)}}\right]^T \psi_R^{(k)} + \left[\frac{\partial \mathbf{E}^{(k)}}{\partial \mathbf{T}^{(k)}}\right]^T \psi_E^{(k)} + \left[\frac{\partial \mathbf{G}^{(k)}}{\partial \mathbf{T}^{(k)}}\right]^T \psi_G^{(k)} + \left[\frac{\partial f}{\partial \mathbf{T}^{(k)}}\right]^T = 0, \tag{3.27}$$

$$\mathbf{D}_{\psi}^{(k)} = \left[\frac{\partial \mathbf{D}^{(k)}}{\partial \mathbf{u}_A^{(k)}}\right]^T \psi_D^{(k)} + \left[\frac{\partial \mathbf{R}^{(k)}}{\partial \mathbf{u}_A^{(k)}}\right]^T \psi_R^{(k)} + \left[\frac{\partial \mathbf{E}^{(k)}}{\partial \mathbf{u}_A^{(k)}}\right]^T \psi_E^{(k)} + \left[\frac{\partial f}{\partial \mathbf{u}_A^{(k)}}\right]^T = 0. \tag{3.28}$$

At the initial conditions ($k = 0$) the adjoint equations are:

$$\mathbf{A}_{\psi}^{(0)} = \left[\frac{\partial \mathbf{A}^{(0)}}{\partial \mathbf{q}^{(0)}}\right]^T \psi_A^{(0)} + \sum_{i=1}^{m_1} \left[\left(\frac{\partial \mathbf{A}^{(i)}}{\partial \dot{\mathbf{q}}^{(i)}}\right) \frac{\partial \dot{\mathbf{q}}^{(i)}}{\partial \mathbf{q}^{(0)}}\right]^T \psi_A^{(i)} + \left[\frac{\partial f}{\partial \mathbf{q}^{(0)}}\right]^T = 0, \tag{3.29}$$

$$\mathbf{G}_{\psi}^{(0)} = \left[\frac{\partial \mathbf{A}^{(0)}}{\partial \mathbf{x}_G^{(0)}}\right]^T \psi_A^{(0)} + \sum_{i=1}^{m_1} \left[\left(\frac{\partial \mathbf{A}^{(i)}}{\partial \dot{\mathbf{x}}_G^{(i)}}\right) \frac{\partial \dot{\mathbf{x}}_G^{(i)}}{\partial \mathbf{x}_G^{(0)}}\right]^T \psi_A^{(i)} + \left[\frac{\partial \mathbf{G}^{(0)}}{\partial \mathbf{x}_G^{(0)}}\right]^T \psi_G^{(0)} + \left[\frac{\partial f}{\partial \mathbf{x}_G^{(0)}}\right]^T = 0, \tag{3.30}$$

$$\mathbf{S}_{\psi}^{(0)} = \left[\frac{\partial \mathbf{D}^{(1)}}{\partial \mathbf{u}_S^{(0)}}\right]^T \psi_D^{(1)} + \left[\frac{\partial \mathbf{L}^{(1)}}{\partial \mathbf{u}_S^{(0)}}\right]^T \psi_L^{(1)} + \left[\frac{\partial \mathbf{S}^{(0)}}{\partial \mathbf{u}_S^{(0)}}\right]^T \psi_S^{(0)} + \sum_{i=1}^{m_2} \left[\left(\frac{\partial \mathbf{S}^{(i)}}{\partial \dot{\mathbf{u}}_S^{(i)}}\right) \frac{\partial \dot{\mathbf{u}}_S^{(i)}}{\partial \mathbf{u}_S^{(0)}}\right]^T \psi_S^{(i)} + \sum_{i=1}^{m_3} \left[\left(\frac{\partial \mathbf{S}^{(i)}}{\partial \ddot{\mathbf{u}}_S^{(i)}}\right) \frac{\partial \ddot{\mathbf{u}}_S^{(i)}}{\partial \mathbf{u}_S^{(0)}}\right]^T \psi_S^{(i)} + \left[\frac{\partial f}{\partial \mathbf{u}_S^{(0)}}\right]^T = 0. \tag{3.31}$$

The lagging of structural displacements in the forward algorithm leads to coupling terms from step $k+1$ in the structural adjoint equations. The coupled adjoint equations are solved using Algorithm 6. Like the steady aeroelastic problem, the information flows in the reverse direction compared to the forward analysis. This is shown in Figure 3.3 for the first two

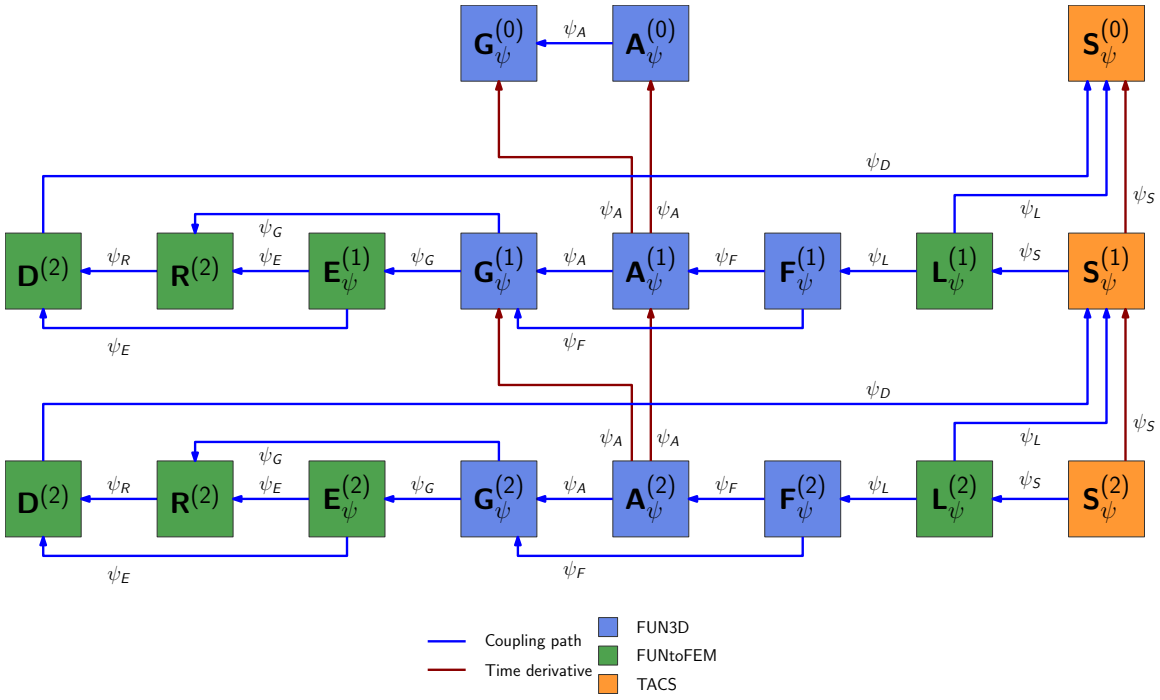


Figure 3.3: Flow of information during the time-accurate adjoint analysis.

time steps. The adjoint procedure starts at the bottom right of the flow chart and evaluates the adjoint residuals from right to left and bottom to top until the initial conditions have been reached.

3.2.3 Adjoint-based Sensitivities

From the adjoint solution, the gradient of the objective function determined by the partial derivative of the Lagrangian with respect to the design variables:

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial \mathbf{x}} = \frac{\partial f}{\partial \mathbf{x}} + \sum_{k=1}^{N} \Bigg[&\left(\psi_D^{(k)}\right)^T \frac{\partial D^{(k)}}{\partial \mathbf{x}} + \left(\psi_R^{(k)}\right)^T \frac{\partial R^{(k)}}{\partial \mathbf{x}} + \left(\psi_E^{(k)}\right)^T \frac{\partial E^{(k)}}{\partial \mathbf{x}} + \left(\psi_G^{(k)}\right)^T \frac{\partial G^{(k)}}{\partial \mathbf{x}} \\ &+ \left(\psi_A^{(k)}\right)^T \frac{\partial A^{(k)}}{\partial \mathbf{x}} + \left(\psi_F^{(k)}\right)^T \frac{\partial F^{(k)}}{\partial \mathbf{x}} + \left(\psi_L^{(k)}\right)^T \frac{\partial L^{(k)}}{\partial \mathbf{x}} + \left(\psi_S^{(k)}\right)^T \frac{\partial S^{(k)}}{\partial \mathbf{x}} \Bigg] \\ &+ \left(\psi_G^{(0)}\right)^T \frac{\partial G^{(0)}}{\partial \mathbf{x}} + \left(\psi_A^{(0)}\right)^T \frac{\partial A^{(0)}}{\partial \mathbf{x}} + \left(\psi_S^{(0)}\right)^T \frac{\partial S^{(0)}}{\partial \mathbf{x}}\end{aligned} \tag{3.32}$$

Like the steady aeroelastic sensitivities, this general expression can be simplified de-

Algorithm 6 Linear block Gauss-Seidel method for the unsteady adjoint solution

Given: k_{max}

$\psi_D^{(k_{max}+1)} \leftarrow 0, \psi_L^{(k_{max}+1)} \leftarrow 0$

$\psi_S^{(k_{max}+1)}, ..., \psi_S^{(k_{max}+m_3)} \leftarrow 0$

$\psi_A^{(k_{max}+1)}, ..., \psi_A^{(k_{max}+m_1)} \leftarrow 0$

for $k = k_{max}$ to 1 **do**

$\psi_S^{(k)} \leftarrow \mathbf{S}_\psi^{(k)} \left(\psi_S^{(k)}, ..., \psi_S^{(k+m_3)}, \psi_D^{(k+1)}, \psi_L^{(k+1)} \right) = 0$ ▷ Structural adjoint eqn.

$\psi_L^{(k)} \leftarrow \mathbf{L}_\psi^{(k)} \left(\psi_L^{(k)}, \psi_S^{(k)} \right) = 0$ ▷ Force transfer adjoint eqn.

$\psi_F^{(k)} \leftarrow \mathbf{F}_\psi^{(k)} \left(\psi_F^{(k)}, \psi_L^{(k)} \right) = 0$ ▷ Force integration adjoint eqn.

$\psi_A^{(k)} \leftarrow \mathbf{A}_\psi^{(k)} \left(\psi_A^{(k)}, ..., \psi_A^{(k+m_1)}, \psi_F^{(k)} \right) = 0$ ▷ Aerodynamic adjoint eqn.

$\psi_G^{(k)} \leftarrow \mathbf{G}_\psi^{(k)} \left(\psi_G^{(k)}, \psi_A^{(k)}, ..., \psi_A^{(k+m_1)}, \psi_F^{(k)} \right) = 0$ ▷ Grid deformation adjoint eqn.

$\psi_E^{(k)} \leftarrow \mathbf{E}_\psi^{(k)} \left(\psi_E^{(k)}, \psi_G^{(k)} \right) = 0$ ▷ Elastic surface deformation adjoint eqn.

$\psi_R^{(k)} \leftarrow \mathbf{R}_\psi^{(k)} \left(\psi_R^{(k)}, \psi_E^{(k)}, \psi_G^{(k)} \right) = 0$ ▷ Rigid adjoint eqn.

$\psi_D^{(k)} \leftarrow \mathbf{D}_\psi^{(k)} \left(\psi_D^{(k)}, \psi_R^{(k)}, \psi_E^{(k)} \right) = 0$ ▷ Displacement transfer adjoint eqn.

end for

$\psi_A^{(0)} \leftarrow \mathbf{A}_\psi^{(0)} \left(\psi_A^{(0)}, ..., \psi_A^{(m_1)} \right) = 0$ ▷ Aerodynamic initial condition adjoint eqn.

$\psi_G^{(0)} \leftarrow \mathbf{G}_\psi^{(0)} \left(\psi_G^{(0)}, \psi_A^{(0)}, ..., \psi_A^{(m_1)} \right) = 0$ ▷ Grid deformation initial condition adjoint eqn.

$\psi_S^{(0)} \leftarrow \mathbf{S}_\psi^{(0)} \left(\psi_S^{(0)}, ..., \psi_S^{(m_3)}, \psi_D^{(1)}, \psi_L^{(1)} \right) = 0$ ▷ Structural initial condition adjoint eqn.

pending on the design variable. For example, the simplified expression for aerodynamic design variables is

$$\frac{\partial \mathscr{L}}{\partial \mathbf{x}} = \frac{\partial f}{\partial \mathbf{x}} + \sum_{k=1}^{N} \left[\left(\psi_A^{(k)} \right)^T \frac{\partial A^{(k)}}{\partial \mathbf{x}} \right] + \left(\psi_A^{(0)} \right)^T \frac{\partial A^{(0)}}{\partial \mathbf{x}}. \tag{3.33}$$

The shape derivatives are split according to the chain rule between terms associated with the initial aerodynamic surface coordinates and the initial structural coordinates:

$$\begin{aligned}
\frac{\partial \mathscr{L}}{\partial \mathbf{x}} = & \left\{ \frac{\partial f}{\partial \mathbf{x}_{A0}} + \sum_{k=1}^{N} \left[\left(\psi_D^{(k)} \right)^T \frac{\partial \mathbf{D}^{(k)}}{\partial \mathbf{x}_{A0}} + \left(\psi_R^{(k)} \right)^T \frac{\partial \mathbf{R}^{(k)}}{\partial \mathbf{x}_{A0}} + \left(\psi_E^{(k)} \right)^T \frac{\partial \mathbf{E}^{(k)}}{\partial \mathbf{x}_{A0}} \right. \right. \\
& \left. \left. + \left(\psi_L^{(k)} \right)^T \frac{\partial \mathbf{L}^{(k)}}{\partial \mathbf{x}_{A0}} + \left(\psi_G^{(k)} \right)^T \frac{\partial \mathbf{G}^{(k)}}{\partial \mathbf{x}_{A0}} \right] + \left(\psi_G^{(0)} \right)^T \frac{\partial \mathbf{G}^{(0)}}{\partial \mathbf{x}_{A0}} \right\} \frac{\partial \mathbf{x}_{A0}}{\partial \mathbf{x}} \\
& + \left\{ \frac{\partial f}{\partial \mathbf{x}_{S0}} + \sum_{k=1}^{N} \left[\left(\psi_D^{(k)} \right)^T \frac{\partial \mathbf{D}^{(k)}}{\partial \mathbf{x}_{S0}} + \left(\psi_L^{(k)} \right)^T \frac{\partial \mathbf{L}^{(k)}}{\partial \mathbf{x}_{S0}} + \left(\psi_S^{(k)} \right)^T \frac{\partial \mathbf{S}^{(k)}}{\partial \mathbf{x}_{S0}} \right] \right. \\
& \left. + \left(\psi_S^{(0)} \right)^T \frac{\partial \mathbf{S}^{(0)}}{\partial \mathbf{x}_{S0}} \right\} \frac{\partial \mathbf{x}_{S0}}{\partial \mathbf{x}}.
\end{aligned} \tag{3.34}$$

To avoid having to save the adjoint solution at every time step, the summations in the Equations 3.32 and 3.34 are accumulated during the reverse time marching in Algorithm 5. For the shape derivatives, those summations including the step zero contribution are multiplied by the design velocities from the shape parameterization after the completion of the algorithm.

3.3 Coupled Solver Validation

3.3.1 Vortex-induced Vibration of a Cylinder

The first verification case for the coupled solver is vortex-induced vibrations of a cylinder. In this problem, the periodic vortex shedding of the cylinder shown in Figure 3.4 causes vibration of the cylinder which is a spring-mass-damper system constrained to only

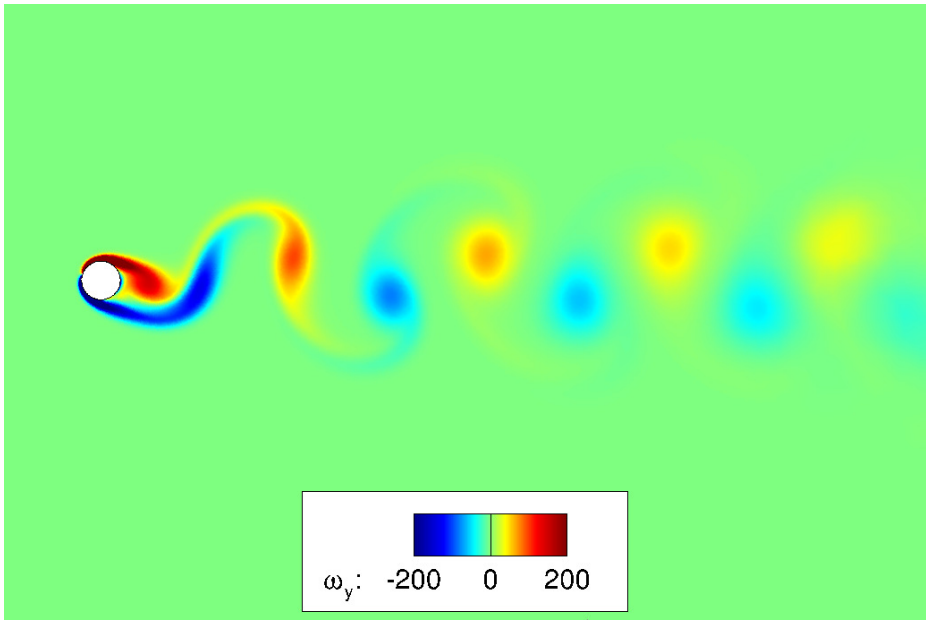


Figure 3.4: Vortex shedding simulated with FUNtoFEM for the vortex-induced vibration case.

move vertically. When the vortex shedding frequency is near the natural frequency of the spring-mass-damper system, the vortex shedding and cylinder vibration can synchronize and create large amplitude vibrations. This synchronization is known as lock-in. Anagnostopoulos and Bearman [129] performed an experimental investigation of an elastically mounted cylinder in crossflow of a water channel to study the vibration behavior in the laminar flow regime. In the experiments, the vibration amplitude and frequency as well as the vortex shedding frequencies were recorded for Reynolds numbers between 90 and 150. The cylinder had a diameter of 1.6 mm and a length of 12 cm. The cylinder's aspect ratio is 75 to approximate a two-dimensional scenario. However, no end plates were mounted on the cylinder, and the measured fixed cylinder shedding frequency is slightly below other two-dimensional studies which suggests that the end effects may have some influence. The leaf spring system on which the cylinder was mounted produced a spring stiffness of 69.48 N/m, and the mass was 35.75 g. The damping ratio of the system in air was measured to be between 1.20×10^{-3} and 1.52×10^{-3}.

The vortex-induced vibration problem was simulated with FUNtoFEM as two-dimensional

Table 3.1: Convergence study for the vortex-induced vibration of a cylinder at $Re_D = 120$.

Mesh	Number of nodes	Time step [s]	h/D
1	29760	0.0008	0.366875
2	88371	0.0008	0.341875
3	291192	0.0008	0.336875
		0.0004	0.338125
4	1046278	0.0008	0.338294
		0.0004	0.339528

model. The incompressible path in FUN3D was applied with the BDF2opt scheme for temporal advancement. The 'rigid+deform' motion option was selected in FUN3D. Because the cylinder is rigid, the full motion is captured with the rigid component from FUNtoFEM, and FUN3D immediately exits the mesh deformation solver because the initial grid deformation residual is approximately machine precision. The structural aspects of the problem are modeled as a spring-mass-damper system governed by

$$m\ddot{z} + c\dot{z} + kz = f_{z,external}. \tag{3.35}$$

where the external force is the total vertical load from the fluid. This equation is discretized with a second-order backwards difference formula (BDF2) and implemented in Python.

A mesh and time step study was performed with FUNtoFEM and is summarized in Table 3.1. The output of interest is the amplitude of oscillation normalized by the diameter of the cylinder. The larger time step, 8.0×10^{-4} s, produced approximately 180 steps per oscillation cycle. Since the mesh 3 amplitude at the larger time step was within 0.5% of the finest mesh at the finer time step, mesh 3 with the larger time step was considered to be sufficiently converged. The simulations that are compared against experiment in the following results are performed with this level of refinement.

The vortex-induced vibration of a cylinder was simulated with Reynolds numbers between 95 and 140, and the results are compared to Anagnostopoulos and Bearman [129]. In the experiments of Anagnostopoulos and Bearman, this Reynolds number range bracketed

the lock-in range. Since the frequencies change with Reynolds number, the time step size was adjusted to maintain approximately 180 time steps per oscillation cycle. Figure 3.5 illustrates how the amplitude of the cylinder oscillations changes as the Reynolds number is increased. The two experimental curves represent different initial conditions where the data in black is collected when the cylinder was released from rest at each Reynolds number, the data in blue was collected when cylinder started from the previous oscillation as the Reynolds number was increased. The red points are the results from the two-dimensional FUNtoFEM runs. All of the data sets show very small amplitudes for Reynolds numbers below 100. Above this point, the amplitude suddenly increases as lock-in occurs. For the experiments from initial oscillations and from rest, this increase occurs around Reynolds numbers of 102 and 104, respectively. For the simulations, the increase occurs at a Reynolds numbers of 108. As the Reynolds number continues to increase, the amplitude of oscillations drops as the lock-in breaks down, and two distinct frequencies appear.

The FUNtoFEM simulations capture the trends of the experimental results but predict the start of lock-in at a higher Reynolds number and lower amplitudes at all Reynolds number in the lock-in range. The peak amplitude predicted by the simulations is h/D =0.41, while the experiments had peak nondimensional amplitudes of 0.60 and 0.53. The biggest difference in physics is that FUNtoFEM is modeling a two-dimensional problem while the experiment setup includes three-dimensional effects at the tip of the cylinder at one end as well as a free surface at the point where the cylinder is submerged in the channel. Anagnostopoulos and Bearman also indicate that the turbulence level in the channel as high as 1%, which is not modeled. These error trends could also be indicative of too much dissipation in simulations due to the method of problem discretization. Since there are differences in modeled physics and sensitivity of the experimental data (13% difference in the experimental peak amplitude), the prediction of the proper trends indicates good agreement between the experiment and the computation.

For further correlation, the vortex-induced vibration case compared directly to FUN3D's

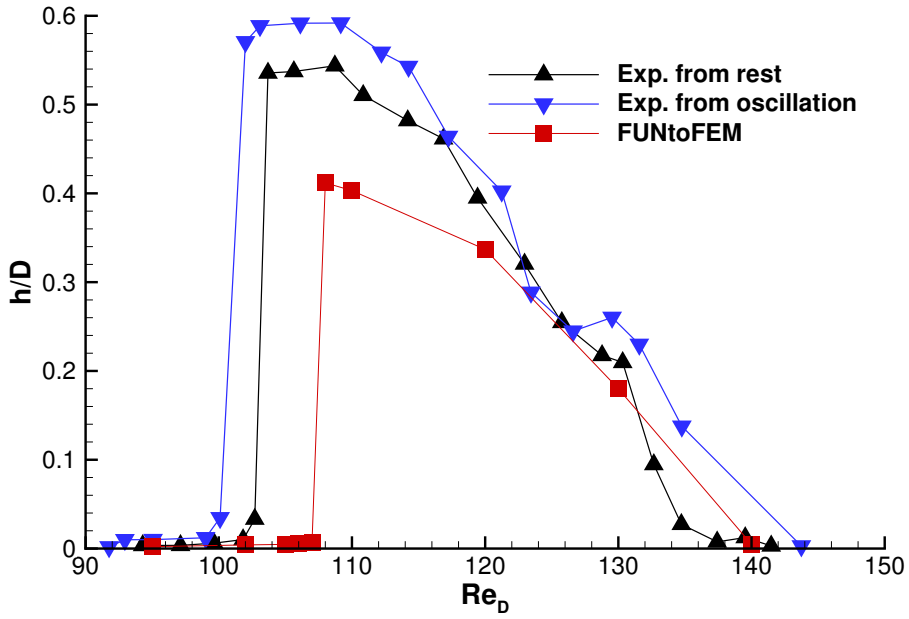
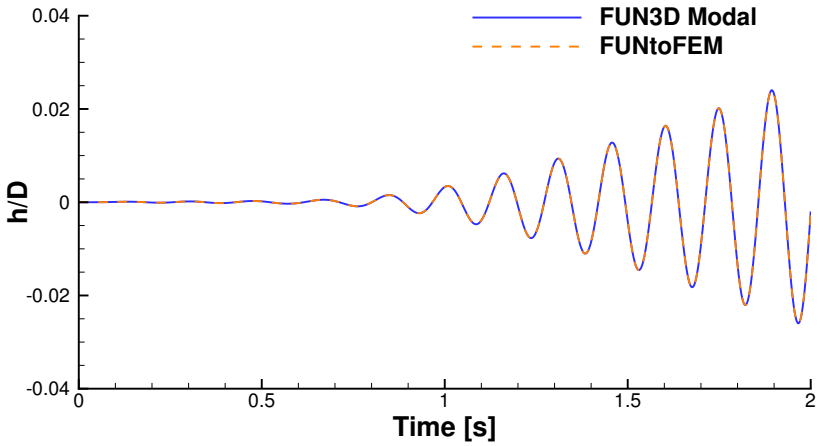


Figure 3.5: Vibration amplitude versus Reynolds number for the vortex-induced vibration case.

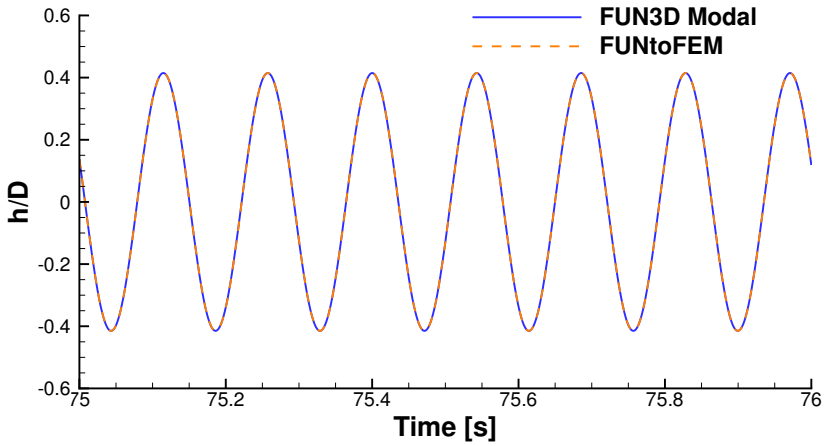
modal solver capability. For this comparison, all of the flow solver options in FUN3D were identical, and both FUNtoFEM and FUN3D's modal structural solver employed a BDF2 temporal discretization. The only difference between the simulations is that the FUN3D modal solver requires projection of the mode shapes onto the CFD surface while FUNtoFEM uses the original structural mesh and the load and displacement transfer. Figure 3.6 compares the displacement predicted by the FUN3D modal solver and FUNtoFEM. At each time step, there is between three and five significant digits of agreement between the simulations. Since FUN3D's modal solver has been verified for aeroelastic analysis with linear structures [99], agreement between FUNtoFEM and FUN3D's modal solver is a good indication that FUNtoFEM is working for the vortex-induced vibration of a cylinder.

3.3.2 Benchmark Supercritical Wing

The next verification case is the Benchmark Supercritical Wing (BSCW) which is an aeroelastic flutter test case from the first and second Aeroelastic Prediction Workshops [130, 131]. The BSCW is a 16x32 inch rectangular wing with no twist or sweep and a SC(2)-



(a) Initial transients.



(b) Periodic state.

Figure 3.6: Comparison of the vortex-induced vibration displacements with FUNtoFEM and FUN3D's modal solver.

0414 supercritical airfoil. Experiments were performed in the NASA Transonic Dynamics Tunnel. The wing was mounted on a splitter plate to place the wing closer to the wind tunnel centerline and reduce the effects of the tunnel wall boundary layer. The wing was mounted in a way such that is can be modeled as approximately rigid with springs at the root midchord that give the wing pitch and plunge degrees of freedom. The specific test case studied here is Case 2 from the second Aeroelastic Prediction Workshop. For this condition, the wind tunnel fluid is R-12 and the Mach number is 0.74.

The BSCW model was again simulated with FUNtoFEM coupled to FUN3D, and the spring-mass-damper integrator from the vortex-induced vibration test case was modified to include a rotational degree of freedom. In the FUN3D setup, the compressible path with Roe's fluxes and the Spalart-Allmaras turbulence model was employed with the BDF2opt temporal scheme and ten pseudo-time subiterations. With ten subiterations, the pseudo-time mean-flow residuals dropped between six and seven orders of magnitude each time step in the BSCW simulations.

The BSCW simulations were performed in three phases. For the first 100 time steps, the wing was held at zero degrees angle of attack, while the initial flow transients propagated away from the wing. Next, the wing was released and critical damping ($\zeta = 1$) was applied to both structural degrees of freedom so that the wing could reach static equilibrium. After 3,000 time steps, static equilibrium has been approximately achieved. At this point, the structural damping was removed, and the pitch degree of freedom was given an instantaneous change in velocity to excite oscillations of the system. These oscillations were allowed to evolve, and the damping of the system was then calculated from approximately 3 to 9 seconds of total simulated time. The total number of time steps to simulate this 9 seconds of data was 45,000. Figure 3.7 gives an example response from this process. The process was performed at four dynamic pressures: 168.8 psf (the experimental flutter point), 158.7 psf (the flutter point of the FUNtoFEM computation), 152.0 psf (the computational flutter point from Chwalowski and Heeg [132], and 135.2 psf (the next high-

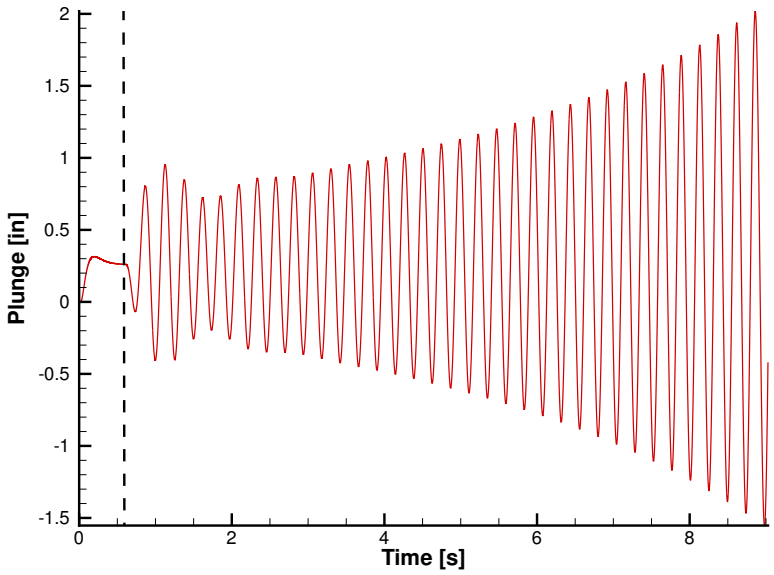


Figure 3.7: Plunge response of the BSCW at $q_\infty = 168.8$ psf (case 2, Mach = 0.74, AOA=0.0). The dashed line represents the time at which the structure is externally excited, and the critical damping is removed.

est dynamic pressure presented by Chwalowski and Heeg). The aerodynamic meshes are the coarse and medium mixed-element, node-centered meshes provided by the Aeroelastic Prediction Workshop which contained 3 million and 9 million nodes, respectively [131, 133].

The results from FUNtoFEM are compared to those from Chwalowski and Heeg [132]. In Chwalowski and Heeg, the FUN3D solver utilized its internal modal structural solver. Table 3.2 highlights the differences in the simulations performed with FUNtoFEM and the FUN3D modal solver. Because the wing was approximated as a rigid structure, the structural dynamics of the system in FUNtoFEM have been represented by spring-mass-damper systems with vertical translation and pitch rotation degrees of freedom. However, the modal solver in FUN3D does not allow the rotational degree of freedom and must represent it as a linearized rotation, i.e., a linear distribution of vertical displacements to approximate a small rotation. This linearized rotation slightly elongates the chord and shears the shape of the wing as it pitches. Another difference is that the modal solver implementation in

FUN3D only permits deforming volume mesh motion, whereas the FUNtoFEM model can utilize the 'rigid+deform' mesh movement option. As in the simulations of the vortex-induced vibration of a cylinder, FUNtoFEM captures the full motion of the system with the rigid motion component and immediately exits the mesh deformation solver. Therefore, the addition of the rigid mesh movement eliminates the computational cost of the CFD volume mesh deformation at every time step. It also maintains constant mesh quality whereas in the deforming version, the mesh quality varies as the mesh moves. The FUN3D modal solver simulations included 200 flow pseudo-time subiterations with the temporal controller set to 0.1 rather than the fixed ten subiterations in the FUNtoFEM analysis. With the temporal error controller, the average number of subiterations was around 60 in the FUN3D simulations; however, the larger number of subiterations typically dropped the mean flow residuals only one order of magnitude more than the six to seven orders of magnitude observed in the ten subiteration cases. Since six to seven orders of magnitude convergence is typically sufficient, the small additional accuracy was deemed not worth the approximately six times cost increase for the additional subiterations. An additional difference between the simulations is found in temporal discretization of the structural solver: the modal solver in FUN3D utilized a second-order predictor-corrector method while the Python spring-mass-damper integrator in the FUNtoFEM solver employs a second-order BDF scheme. With approximately 1,200 time steps per oscillation, the difference in temporal error between these two second-order schemes is not expected to be a major source of error. This was demonstrated for the BSCW by Chwalowski et al. [134] who concluded that there was no significant difference whether the predictor-corrector or BDF scheme was selected for structural time integration.

Table 3.3 compares the integrated loads predicted at static equilibrium by the two simulations at the experimental flutter point. Although the drag predictions agree, there are 1.5% and 1.2% differences in the lift and moment coefficients, respectively. Because these are converged steady results, the structural time integration and number of pseudo-time

Table 3.2: Differences between FUNtoFEM and FUN3D model solver simulation options from Chwalowski and Heeg [132] for modeling of the Benchmark Supercritical Wing

	FUNtoFEM	FUN3D modal solver [132]
Pitch degree of freedom	full rotation	linearized rotation
Mesh motion	rigid + deform	deform
Number of pseudo-time subiterations	10	200 + error controller
Structural time integration	BDF2	predictor-corrector

Table 3.3: Predicted static aeroelastic loads for the BSCW case 2 at $q_\infty = 168.8$ psf with the medium Aeroelastic Prediction Workshop mesh.

	C_L	C_D	C_{My}
FUNtoFEM	0.0949	0.0117	-0.0845
FUN3D modal solver [132]	0.0935	0.0117	-0.0855

steps are not possible sources of the discrepancy. The angle of attack at static equilibrium is -1.0567 degrees in the FUNtoFEM simulation and -1.0372 degrees in the FUN3D solution which means the wing elongation is about 0.02% in the modal solver and is likely not significant. By process of elimination, these differences in results are potentially due to mesh changes during the grid deformation process or the shearing of the airfoils from the linearized rotation. These sources of differences have implications for the dynamic simulations because plunging and pitching amplitudes are larger in the dynamic responses, so the effects of the linearized rotation and mesh deformation will be more significant.

Figures 3.8 and 3.9 compare the predicted frequency and damping ratio of the dynamic response at various dynamic pressures. Within each simulation tool, the coarse and medium mesh results indicate little variation, implying consistency in the results with respect to spatial resolution. In Chwalowski and Heeg's results [132] a fine mesh with 27 million nodes also produced comparable predictions to the coarse and medium meshes. For the frequencies, the simulation results have relative differences of less than 0.5% between the modal solver and FUNtoFEM. The FUNtoFEM frequencies are the larger of the simulation results which represent a small shift towards the experimental frequency of 4.3 Hz at 168.8 psf.

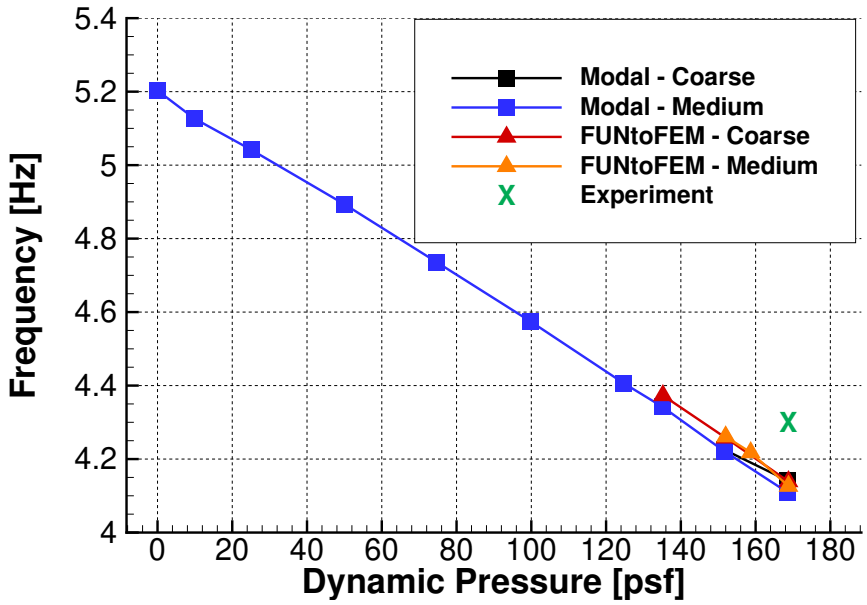


Figure 3.8: Frequency of the dynamic response of the BSCW at various dynamic pressures (case 2, Mach = 0.74, AOA=0.0).

The FUNtoFEM simulations predict higher damping than the modal solver results. At the experimental flutter point, the FUNtoFEM simulations predict about 70% as much negative damping as the modal solver results. The flutter dynamic pressure from FUNtoFEM is 158.7 psf which has a 6.0% relative error with respect to the experiment flutter point. The relative error is 10.0% for the modal solver.

Figures 3.10 and 3.11 compare the mean pressure distributions of the computational and experimental results at their respective flutter dynamic pressures. On the upper surface at both span locations, the FUN3D and FUNtoFEM results are within the 99% bound of the experiment over most of the chord except near $x/c = 0.85\%$. On the lower surface, the computational results have a stronger mean suction peak near the leading edge and higher mean pressure in the cusp of the supercritical airfoil ($x/c > 0.75$). This was a consistent trend in the computational results across the Aeroelastic Prediction Workshop participants. The FUN3D and FUNtoFEM mean pressure distributions are in good agreement. The largest discrepancy is the behavior of the suction peak on the lower surface ($x/c = 0.05$ to $x/c = 0.15$) where the maximum difference in mean pressure coefficient is 0.04. It is

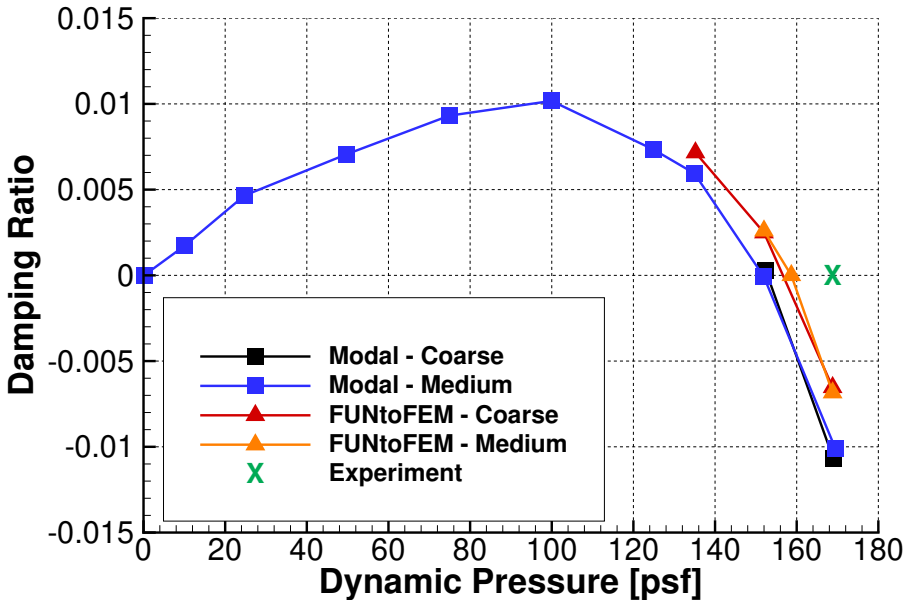


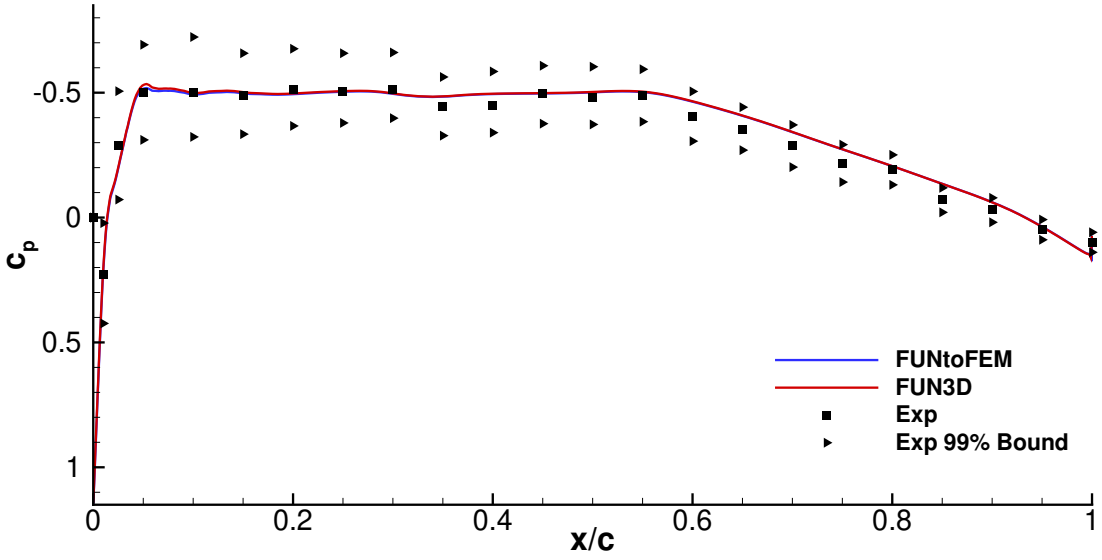
Figure 3.9: Damping ratio of the dynamic response of the BSCW at various dynamic pressures (case 2, Mach = 0.74, AOA=0.0).

hypothesized that this difference is caused by the difference in the representation of the rotational degree of freedom. Figure 3.12 depicts the difference in airfoil shape at negative two degrees. In the FUNtoFEM flutter simulation, the airfoil oscillates from approximately zero to negative two degrees. Although the cross-sections near the trailing edge are nearly identical, the linearly rotated airfoil (FUN3D) has some differences in the lower surface at the leading edge. The slightly higher surface curvature on the lower surface should lead to marginally higher suction as is observed in Figures 3.10b and 3.11b.

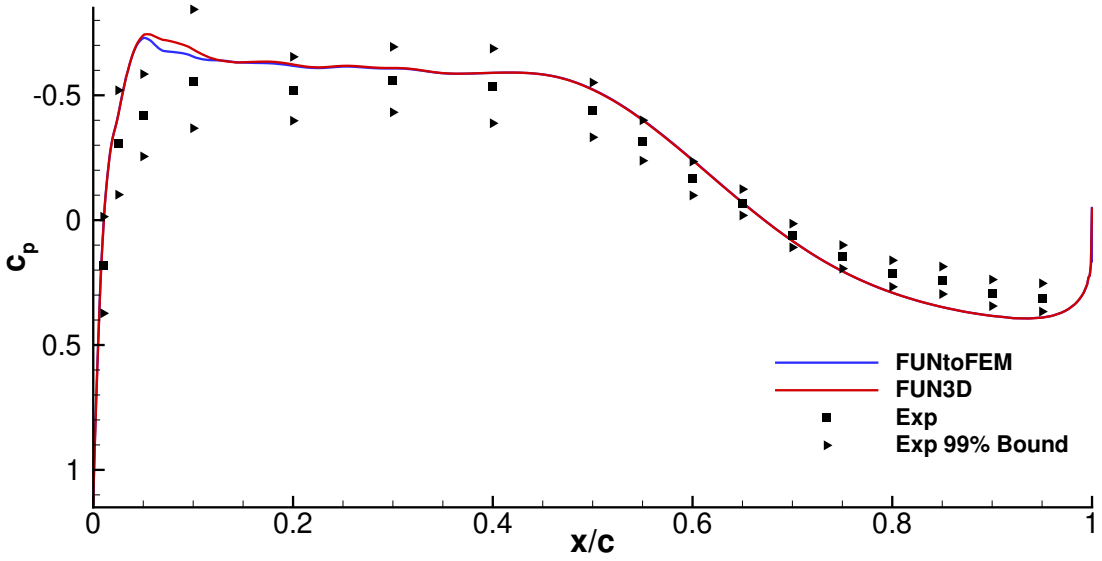
Overall, FUNtoFEM predicted trends similar to the FUN3D modal solver for the oscillation frequency, damping ratio, and mean pressure distributions, but FUNtoFEM predicted a slightly higher damping, resulting in a 6.7 psf increase in flutter dynamic pressure.

3.4 Coupled Sensitivity Verification

The implementation of the coupled adjoint has been verified by comparing adjoint-based derivatives against those calculated by the complex step method. As described in Section 1.3, the complex step method produces sensitivities that are as accurate as the func-

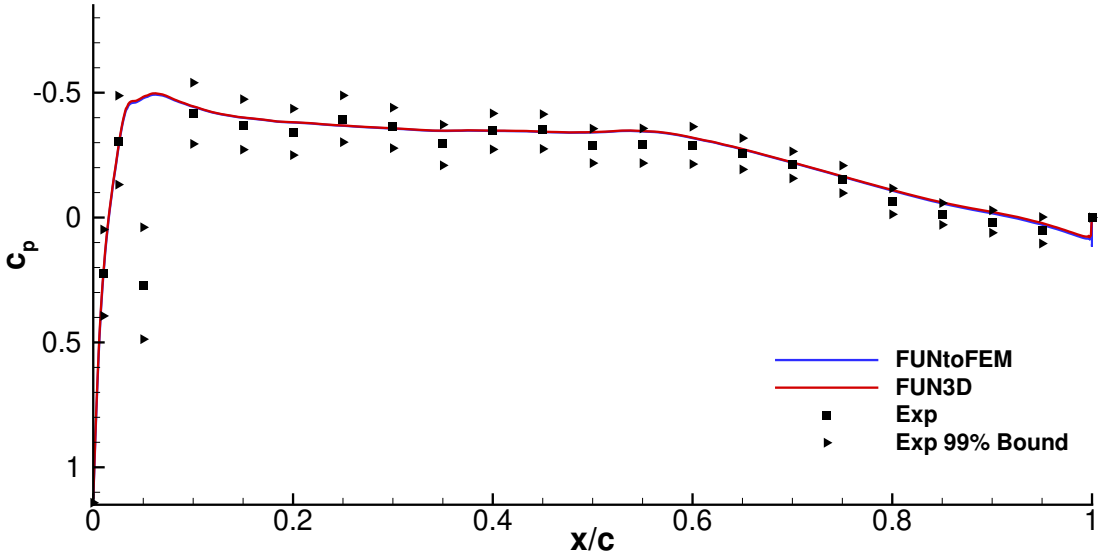


(a) Upper Surface, 60%.

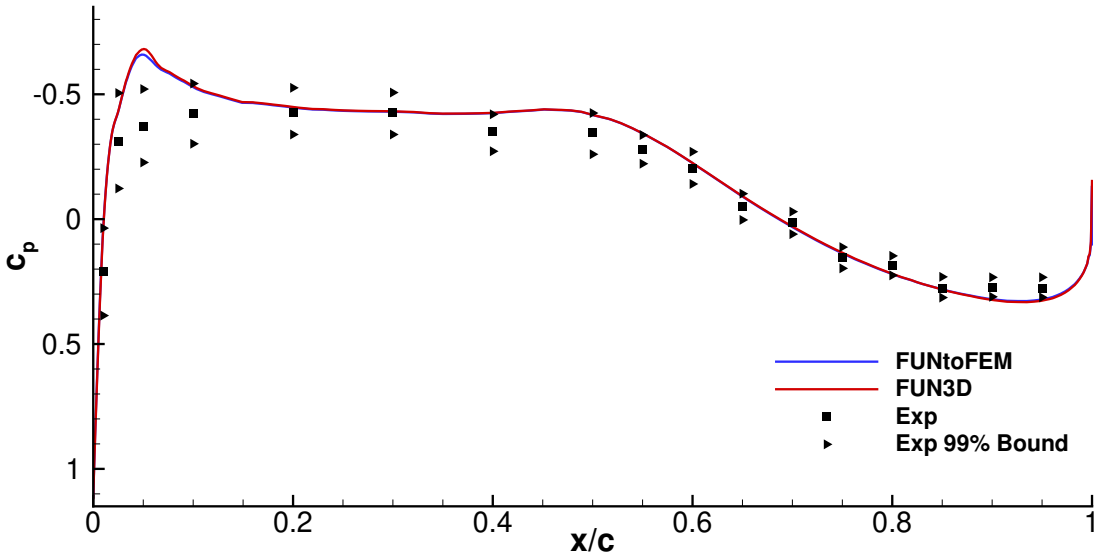


(b) Lower Surface, 60%.

Figure 3.10: Mean pressure distributions at the flutter dynamic pressure and 60% span (BSCW case 2, Mach = 0.74, AOA=0.0).

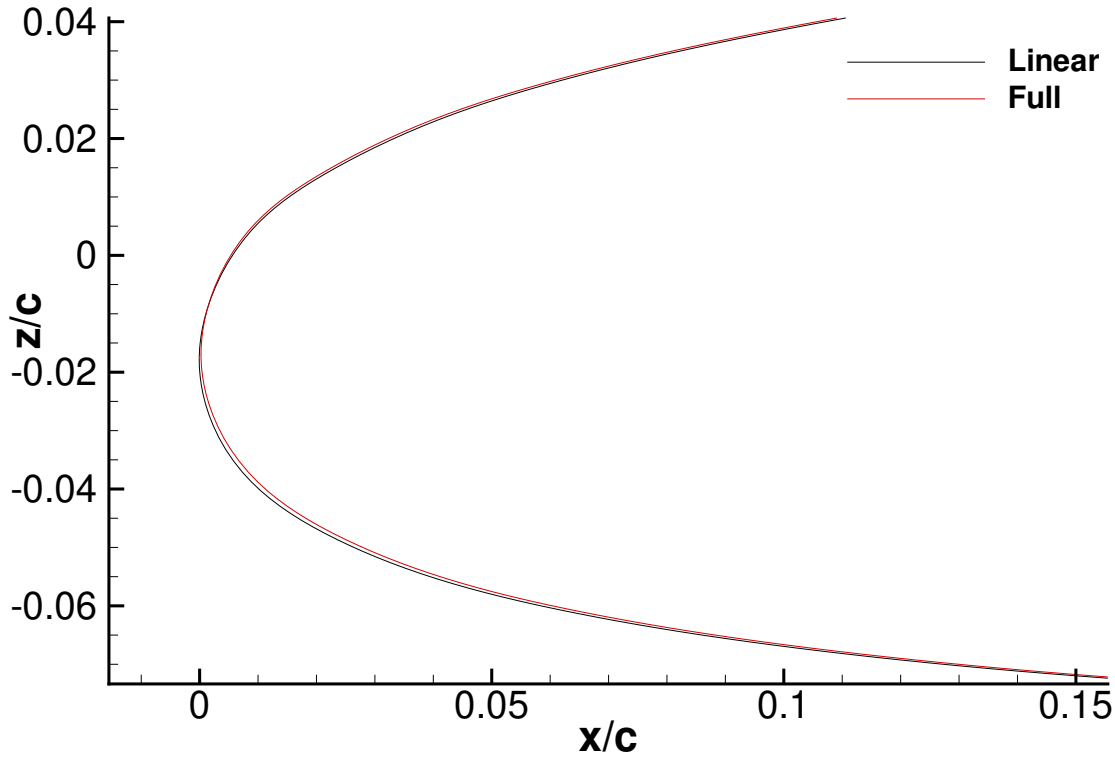


(a) Upper Surface, 95%.

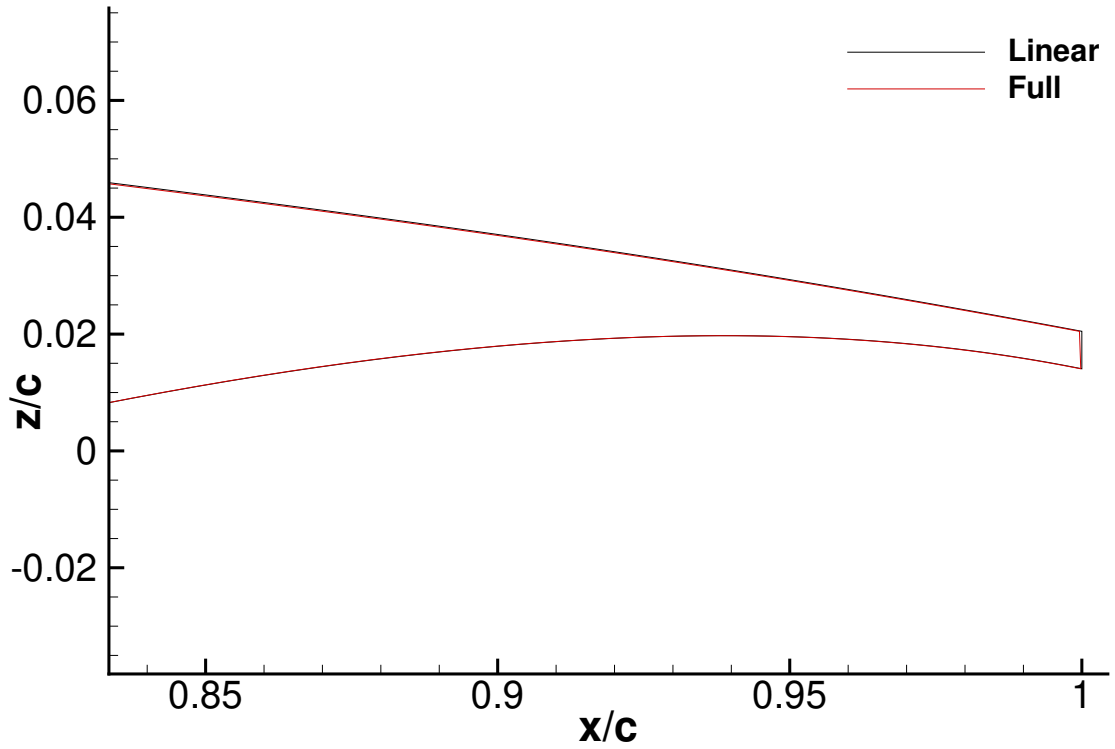


(b) Lower Surface, 95%.

Figure 3.11: Mean pressure distributions at the flutter dynamic pressure and 95% span (BSCW case 2, Mach = 0.74, AOA=0.0).



(a) Leading Edge.



(b) Trailing Edge.

Figure 3.12: Comparison of the full and linearized rotation of the BSCW at -2.0 degrees.

Table 3.4: Steady coupled adjoint verification comparing derivatives based on the adjoint and complex step approaches.

	Structural Thickness	Angle of Attack	Shape Variable
Lift Adjoint	-0.00069148366524	30.561349741748	6.357127008097**8**
Lift Complex	-0.00069148366510	30.561349741749	6.357127008096**7**
KS Failure Adjoint	-1.513782969**2**$\times 10^{-5}$	0.0085284874047**5**	0.0074682888033**8**
KS Failure Complex	-1.513782968**5**$\times 10^{-5}$	0.0085284874047**9**	0.0074682888033**7**

tion calculation, and therefore this method is a good basis for verifying the adjoint implementation. For all of the complex step results in this chapter, the complex step size was 1.0×10^{-30}.

3.4.1 Static Aeroelastic Sensitivities

The static aeroelastic sensitivities from the adjoint and complex step methods are given in Table 3.4 for the undeformed Common Research Model (uCRM) configuration [66]. The uCRM is a representative civilian transonic transport model with a wing box structure represented by linear shell elements in TACS. Both an aerodynamic function and a structural function are studied. The aerodynamic function is the lift generated by the wing. The structural function is the Kreisselmeier-Steinhauser (KS) [135–137] aggregation of von Mises stress which is a differentiable approximation of the maximum stress in the structure. The design variables in the sensitivity verification include the thickness of a panel of the wing box structure, the aerodynamic angle of attack, and a shape variable that controls the x coordinate location of a parameterization control point in MASSOUD. The real and complex forward analyses and the adjoint problems for both functions have been converged to machine precision for this comparison. In the table, the derivatives have at least nine digits of agreement between the adjoint and complex step methods which verifies the steady adjoint implementation.

Table 3.5: Vortex induced vibration sensitivities of energy harvested.

	Angle of Attack	Stiffness	Damping Coefficient
Adjoint	$6.7947745142 \times 10^{-8}$	$4.2440343616 \times 10^{-9}$	$8.40211347566 \times 10^{-8}$
Complex	$6.7947745161 \times 10^{-8}$	$4.2440343619 \times 10^{-9}$	$8.40211347566 \times 10^{-8}$

3.4.2 Time-accurate Aeroelastic Sensitivities

To verify the time-accurate aeroelastic sensitivities for the range of problems studied in this research, two different problems are presented. The first case is the vortex-induced vibration of a cylinder represented by a modal structure with the combined rigid and deforming motion option in FUNtoFEM and FUN3D. The second case is the uCRM case utilized in the steady aeroelastic sensitivity verification. This case has a linear finite element model in TACS, deform-only CFD volume mesh motion, and MASSOUD-based shape parameterization.

Table 3.5 gives the adjoint and complex finite-difference-based sensitivities for the vortex-induced vibrations of a cylinder. The spring-mass-damper system of the vortex induced vibration system is represented with an adjoint-enabled modal solver written in Python. In the simulation, the structure begins with an initial offset of -0.1 diameter, and the flow initializes from a uniform free-stream. The objective function is the time-integrated energy extracted from the system by the damper which will be optimized in Section 5.1. The three design variables are the angle of attack, the stiffness of the spring, and the damping coefficient. The derivatives of the function with respect to all three design variables agree to at least ten digits between the adjoint and complex step methods. The magnitude of the derivatives are small because the time-accurate simulation is only ten time steps with 200 subiterations per step to converge the pseudo-time residuals to machine precision at each time step; however, this does verify that the mathematics have been implemented properly.

The time-accurate aeroelastic sensitivities for the uCRM are compared in Table 3.6. For

Table 3.6: Comparison of unsteady aeroelastic derivatives calculated by the adjoint and complex step approaches with linear shell elements.

	Structural Thickness	Angle of Attack
Lift Adjoint:	$8.28439764813\times10^{-7}$	$7.83592732405\times10^{-6}$
Lift Complex:	$8.28439764814\times10^{-7}$	$7.83592732405\times10^{-6}$
KS Failure Adjoint:	-0.0306817554094	$1.34592942323\times10^{-7}$
KS Failure Complex:	-0.0306817554094	$1.34592942323\times10^{-7}$
	Shape Variable	
Lift Adjoint:	$2.4092367343\times10^{-5}$	
Lift Complex:	$2.4092367325\times10^{-5}$	
KS Failure Adjoint:	$-1.29787578524\times10^{-3}$	
KS Failure Complex:	$-1.29787578524\times10^{-3}$	

the time-accurate verification, the flow again initializes from a uniform free-stream, but the structure is at rest. Ten time steps are simulated with 200 flow subiterations to converge the pseudo-time stepping to machine precision at each time step. Like the steady aeroelastic sensitivities, the table includes the sensitivities for the lift and KS failure functions with respect to a structural thickness variable, angle of attack, and a MASSOUD shape variable. The lift function is the value at the final time step, and the KS failure function is aggregated over the duration of the simulation. In the table, the adjoint and complex step sensitivities agree to at least ten significant digits. Like the vortex-induced vibration case, the magnitude of the derivatives is small, particularly the cross-discipline derivatives, because there are only ten time steps.

3.4.3 Remote Procedure Call Verification

The final verification of solver implementation is the remote procedure call framework based on Hermes [123]. Since the direct Python coupling version of the solver has been verified in the previous sections, the Hermes-based version is verified by ensuring that it obtains the same answers.

The study was made with the uCRM wing for both steady and unsteady analysis. Ta-

Table 3.7: Function values for steady aeroelastic analysis calculated with remote procedure call implementation.

	Lift	KS Failure
Direct Python - complex	123.630842869	0.5740689110**78**
Direct Python - real	123.630842860	0.5740689110**109**
Hermes - real	123.630842860	0.5740689110**109**

Table 3.8: Steady aeroelastic sensitivities calculated with the remote procedure call implementation.

		Panel thickness 0	Angle of attack
Lift	Direct Python - complex	48.770483361	23.041073108
	Direct Python - real	48.770483364	23.041073106
	Hermes - real	48.770483364	23.041073106
KS failure	Direct Python - complex	-4.6136009**9686**	0.0755157643**653**
	Direct Python - real	-4.6136009**8971**	0.0755157643**636**
	Hermes - real	-4.6136009**8971**	0.0755157643**636**

bles 3.7 and 3.8 compare the Hermes-based function values and sensitivities for the steady aeroelastic analysis. The direct Python and Hermes-based values agree exactly. This is expected since the different version are doing identical calculation, except that the Hermes-based implementation has an extra layer of communication to transfer data between the client and the servers. Tables 3.7 and 3.8 show the Hermes-based function values and sensitivities for the time-accurate aeroelastic analysis which also agree exactly with the direct Python implementation. These results indicate that the Hermes client-server model has been implemented properly. While the Hermes-based remote procedure calls allow the aeroelastic analysis to be performed with different components on separate computers, there is a slight computational performance loss due to the extra communication overhead. The analysis is about 2-5% slower in the remote-procedure call version when all the servers and the client are run on the same computer.

Table 3.9: Function values for time-accurate aeroelastic analysis calculated with the remote procedure call implementation.

	Lift	KS Failure
Direct Python - complex	0.000192515603246	0.0259753355763
Direct Python - real	0.000192515603246	0.0259753355763
Hermes - real	0.000192515603246	0.0259753355763

Table 3.10: Time-accurate aeroelastic sensitivities calculated with the remote procedure call implementation.

		Panel thickness 0	Angle of attack
Lift	Direct Python - complex	-2.00547262678$\times 10^{-5}$	2.89636955472
	Direct Python - real	-2.00547262677$\times 10^{-5}$	2.89636955472
	Hermes - real	-2.00547262677$\times 10^{-5}$	2.89636955472
KS failure	Direct Python - complex	-0.0695002469313	5.1745426714
	Direct Python - real	-0.0695002469313	5.1745426714
	Hermes - real	-0.0695002469313	5.1745426714

CHAPTER 4

AEROELASTIC CONSTRAINTS IN THE TIME DOMAIN

4.1 Flutter Constraints

4.1.1 Types of Flutter Constraints

Flutter constraints are important in nearly all aircraft designs. In the current design process, the final step for all aircraft is flight testing. During flight tests, vehicle characteristics are checked so that the vehicle remains flutter-free everywhere within the flight envelope [138]. For flutter testing, the structure is excited during flight in some manner such as control surface movement, atmospheric turbulence, or inertial exciters. The response to the excitation is measured and analyzed to determine its frequency and damping over the flight envelope. The flight tests are conducted below the flutter speed in a flutter test for the safety of the crew since flutter can lead to catastrophic failure of the vehicle.

For computational models, the safety concern is not present; therefore, the flight conditions can be examined at or above the flutter point. One method of computational flutter constraint that would be analogous to flight test would be a flutter clearance approach where the flight conditions in the flutter simulation(s) are fixed with the constraint that the damping be sufficiently high to ensure safe operation. Another option is a flutter identification approach where the constraint would be that the minimum damping at the flutter condition is zero, and the parameter(s) that control the flutter condition are included as design variables. Limits are set on the design variables that control the flutter condition to ensure that it is safely outside of the flight envelope, e.g., setting a lower bound on the flutter dynamic pressure.

The flutter point is a set of conditions where the minimum damping is zero; however, Figure 3.9 shows that there are multiple conditions of zero damping: $q = 0$ and $q = q_{flutter}$.

One disadvantage of gradient-based optimization is that it will converge to a single local minimum rather than find all of the zero damping points. If the dynamic pressure in a flutter identification optimization is lower than that of the maximum damping due to poor initial condition selection, the optimizer would converge to the local minimum at $q = 0$. While technically not a failure of the optimization algorithm, it is not the desired result. Therefore, some initial knowledge of the damping is necessary to choose an appropriate initial condition. Even with good initial conditions, the flutter identification process may not be robust for large design changes that could push the optimization into the $q = 0$ local minimum. On the other hand, the flutter clearance approach is not as sensitive to initial conditions since the flow conditions are fixed, but it does not provide as much information since the actual flutter condition is not found. That is, the clearance condition confirms that the vehicle is not fluttering, but an exact margin of safety is not known. One potential pitfall of the flutter clearance approach is missing hump modes due a lack of complete coverage of the flight envelope. Hump modes are flutter modes that are unstable for a small range of velocities but are stable above and below this range. Expanding the velocity range results in a large number of time-accurate flutter simulations that will greatly increase the cost of each design cycle.

There are only a few examples of time-accurate optimizations considering flutter in the literature. Mani and Mavriplis [79] as well as Palaniappan et al. [70] minimized the displacement of the final time steps of an initially fluttering airfoil. This essentially maximizes the aerodynamic damping and/or stiffness without directly calculating them. While an effective optimization to verify that their adjoint implementation works, in a more complex design problem it would not sufficient to maximize the damping and/or stiffness because it could lead to overdesigned aircraft. The other set of examples of CFD-based flutter constraints come from Zhang et al. [89–91]. They have demonstrated two methods. The first is a minimization of the square of the lift coefficient which is similar to the optimizations of Mani and Mavriplis and Palaniappan et al. in that it minimizes the response of the system,

which acts as a maximization of the aerodynamic damping. Their other method of flutter constraint is computed from a Hilbert transformation based envelope function that will be explored later in this chapter. Essentially, the envelope function bounds the oscillations in the system's response, and the growth or decay of this bounding function can be used as a representation of the behavior of the oscillations and overall stability. Zhang et al. have applied this method to optimization problems where the objective is flutter point identification (driving the Hilbert transform estimated damping to zero) [90], and they applied it as a constraint in weight minimization [91] where the constraint drove the damping to zero in the optimized design.

Regardless of the choice between flutter clearance and flutter identification, the optimizer must have some representative measure of the damping and its derivative. In the following sections, common damping identification methods, including the Hilbert transformation of Zhang et al., will be described and compared in the context of CFD-based optimization. Desirable characteristics for CFD-based optimization are that the method is robust and differentiable, can handle both positive and negative damping, and can be used in automated manner. Ideally, the computational time per simulation would be minimal; therefore, the comparisons are performed with relatively short time signals that would be typical of CFD simulations.

4.1.2 Damping Calculation Methods

Log Decrement Method

One of the most common methods for estimating damping is the log decrement method. For the method, the log decrement is the natural log of the ratio of two peaks in the signal defined by

$$\delta = \frac{1}{n} \ln \left(\frac{z(t)}{z(t+nT)} \right), \tag{4.1}$$

where n is an integer number of peaks and T is the period of the signal. The damping ratio

is then computed as

$$\zeta = \frac{1}{\sqrt{1+\left(\frac{2\pi}{\delta}\right)^2}}. \tag{4.2}$$

The log decrement method is common selected in analysis for its simplicity. Since it requires only two maxima, the method is applicable to short duration signals which is typical in CFD-based analysis due to computational cost. Technically, the method is only applicable for a single component/frequency signal. Although the log decrement method is not robust enough to consistently handle signals with multiple components, it can sometimes be applied to such cases. For example, given the signal

$$z(t) = 5e^{-t}\sin(50t) - 2.5e^{-0.75t}\sin(45t), \tag{4.3}$$

the damping estimated by the two peaks indicated in Figure 4.1 is 0.965 which is within 5% of the damping of the first component of the signal which has the large amplitude. The lower predicted damping is likely related to the lower damping on the smaller amplitude component of the signal. To apply the log decrement method to a similar signal to this one, the peaks must be intelligently selected. The marked peaks in Figure 4.1 were chosen as by observing the peak pattern and selecting peaks that were the highest local peak in the pattern, i.e., the ones where the local maxima immediately to the left and right are lower magnitude. While the process works in this case, it does not work in general as will be demonstrated in later examples.

Envelope Functions - Hilbert Transformations

One set of methods similar to the log decrement method is the use of envelope functions. With these methods an envelope function enclosing the response and the behavior of the envelope is utilized to estimate the overall stability of the system. The simplest form of an envelope is a discrete envelope formed from connection of the successive local maxima and minima. This is similar to the peak-based analysis of the log decrement method. A

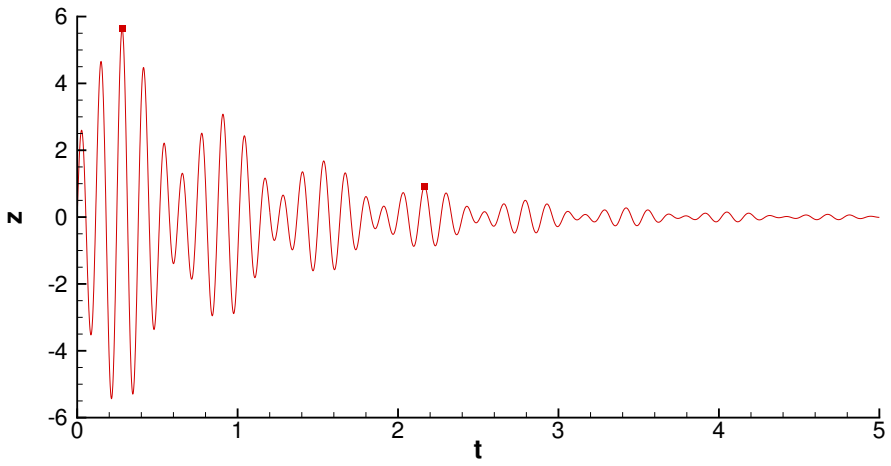


Figure 4.1: Multicomponent signal given by Equation 4.3.

more sophisticated approach is the use a Hilbert transformation which defines the envelope function at every time instance of the original signal. Related to the Fourier transform, the Hilbert transformation adds a 90° phase shift to every Fourier component of a function, and its output is the harmonic conjugate of the original signal. Given a signal $z(t)$, the Hilbert transform is defined as

$$z_v(t) = \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{z(\tau)}{t-\tau}d\tau. \tag{4.4}$$

Following the description of Zhang et al. [90], the signal can then represented as a complex value function given by the harmonic conjugate pair

$$y(t) = z(t) + iz_v(t) = A(t)e^{i\theta(t)}, \tag{4.5}$$

where $A(t)$ is the instantaneous envelope amplitude and $\theta(t)$ is the instantaneous phase of the signal. The instantaneous frequency of the signal is then the rate of change of the phase

$$\omega(t) = \frac{d\theta(t)}{dt}. \tag{4.6}$$

Assuming the signal is an exponentially decaying sinusoid, the instantaneous envelope

amplitude can be expressed as

$$A(t) = e^{-\zeta(t)\omega(t)t}. \tag{4.7}$$

Now an approximation is made that the damping ratio and instantaneous frequency are approximately constant. This is not strictly valid for Hilbert transformation when the signal is not periodic or not infinitely long (the integral in Equation 4.4 is defined for all time), but this permits the damping ratio to be found by taking the natural log of Equation 4.7:

$$\zeta(t) = -\frac{1}{\omega}\frac{d\log(A)}{dt}. \tag{4.8}$$

From this definition of damping ratio, Zhang et al. then form a flutter constraint as

$$f = \int_0^T \zeta(t)^2 dt, \qquad \text{for} \quad \zeta(t) < 0. \tag{4.9}$$

With the $\zeta(t) < 0$ condition, driving this constraint expression to zero suggests that there is no growth in the signal, i.e. the system is not fluttering. If the condition were removed, this would approximate the zero damping condition.

To demonstrate the Hilbert transformation damping calculation, the method is applied the following signal:

$$z(t) = 5e^{-t}\sin(50t), \qquad t \in [0,2]. \tag{4.10}$$

Figure 4.2a shows this signal and the instantaneous envelope amplitude. Because the signal is of finite length, there is some error in the instantaneous envelope amplitude compared to the analytic envelope, $\pm 5e^{t}$. Figure 4.2b illustrates the instantaneous frequency found from Equation 4.6 and damping ratio from Equation 4.8. Again the frequency is not a constant value of 50 due to numerical error from the finite length of the signal. Since the instantaneous damping is calculated from the instantaneous envelope and frequency, there is also error in the damping particularly near the beginning and end of the time range. The

expected value is 1/50 but the value reaches as low as -6.3 at the end of the time history. Using a similar method to Zhang et al. to find a single value of the damping ratio, the mean of the positive damping ratios is 0.193 which has an error of 3.5% compared to 0.02.

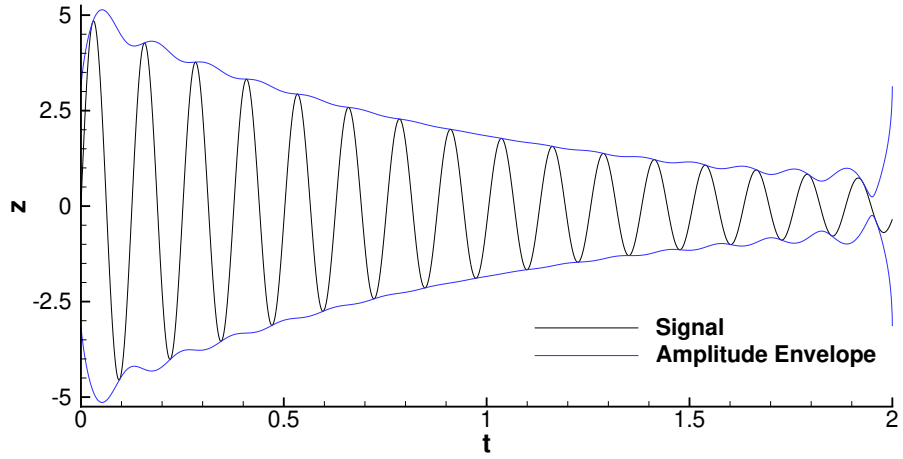
More sophisticated methods have been developed within the realm of envelope functions such as improved Hilbert transforms [139]. However, these more complex methods are not explored here because they all rely on the envelope being representative of the stability of signal. If a small amplitude component of a signal is growing while the initially larger signal component is decaying, it would require a significant time window before the growing signal dominates the overall response and is noticeable in the envelope function. This is not ideal from CFD-based optimization where the length of the simulation should be a short as possible.

Half-power Bandwidth Method

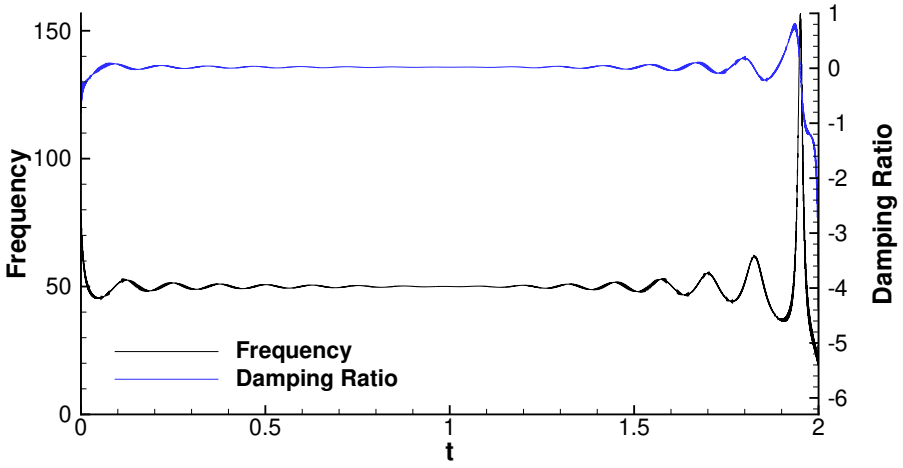
Another common method used in the experimental materials and structures community is the half-power bandwidth method [140–142]. After the signal is converted to the frequency domain, a decaying signal will appear with a peak at the frequency of the decaying signal, ω^*. The width of the peak at the half-power points, ω_1 and ω_2, can be related to the decay of the signal by

$$\zeta = \frac{1}{2}\frac{\omega_2 - \omega_1}{\omega^*}, \tag{4.11}$$

where the half-power points have an amplitude smaller than the peak by $\frac{1}{\sqrt{2}}$. While robust to noise and able to handle multiple signal components, this method is difficult to automate because once the signal is transformed into the frequency domain, it cannot be ascertained whether the oscillation amplitude is growing or decaying. For example, given two signals, $z(t) = 4.5e^{-t}\sin(2\pi 10.0t)$ and $z(t) = 0.05e^{t}\sin(2\pi 10.0t)$ for the time range $[0.0, 4.5]$ in Figure 4.3a, the power spectrums are identical in Figure 4.3b. For simple signals like the ones in this example, the growing signal is easy to identify, and the sign of the damping can be changed to negative. However, when there are multiple components to the signal, it

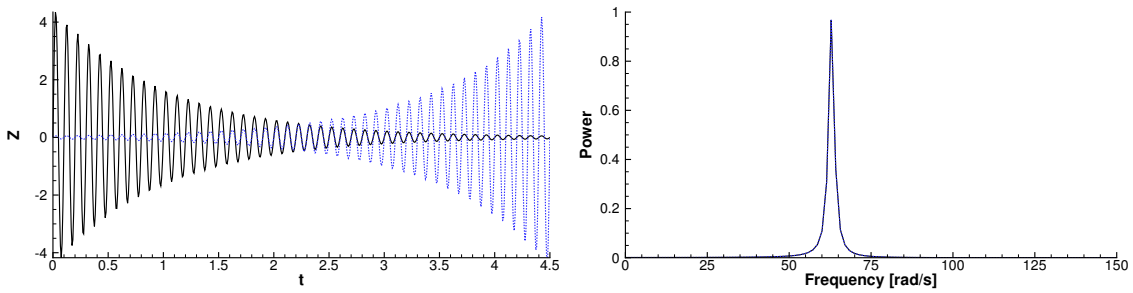


(a) Signal and instantaneous envelope.



(b) Instantaneous frequency and damping ratio.

Figure 4.2: Hilbert transformation example.



(a) Time domain signals. (b) Power spectrums.

Figure 4.3: Half-power example with a growing and decaying signal.

is not as straight forward.

Prony Methods - the Matrix Pencil Method

Rather than deconstruct the signal into constant amplitude sinusoids, Prony series methods decompose the signal into damped sinusoids:

$$z(t) = \sum_{k=1}^{\infty} c_k e^{s_k t}, \tag{4.12}$$

where c_k is a complex number related to the amplitude ($a_k = \text{mod}(c_k)$) and phase ($\phi_k = \arg(c_k)$), and s_k is a complex number which encodes the damping and frequency. Given a set of evenly spaced samples, the typical process to solve for the values of c_k and s_k involves solving a linear equation system, a root finding problem, and then a second linear equation system. One Prony series method of interest is called the matrix pencil method [143, 144]. The matrix pencil method embeds a noise term, $w(n)$, in the decomposition of the signal. The discrete representation of the signal is

$$z_n = \sum_{k=1}^{M} c_k e^{s_k n} + w(n), \qquad n = 0, 1, \ldots, N-1. \tag{4.13}$$

where M is the model order, and now the complex exponent coefficient includes the uniform

time step, $s_k = (\alpha_k + i\omega_k)\Delta t$. Based on this decomposition of the signal, the damping is found with the technique of Sarkar and Pereira [145] outlined in Algorithm 7. The optional first step which subsamples the signal down to N samples and reduces the problem size for the eigenvalue and singular value decomposition steps later in the algorithm. Next, the pencil parameter L is set. This parameter is an important factor that controls the noise filtering and is typically selected to be between $N/3$ and $N/2$. After the pencil parameter is chosen, the Hankel matrix Y is created

$$\mathbf{Y} = \begin{bmatrix} y_0 & y_1 & \cdots & y_{L+1} \\ y_1 & y_2 & \cdots & y_{L+2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N-L-1} & y_{N-L} & \cdots & y_{N-1} \end{bmatrix}. \tag{4.14}$$

Then the singular value decomposition of $\mathbf{Y}$ is found. The model order, M in Equation 4.13 is the number of singular values above a specified tolerance. The first M right singular vectors are kept in $\hat{\mathbf{V}}$, and the remaining are discarded as noise. From these filtered right singular vectors, two subsets are formed: $\hat{\mathbf{V}}_1$ which is the rows 1 through L and $\hat{\mathbf{V}}_2$ which is rows 2 through $L+1$. The matrix $\mathbf{A}$ is formed as the product of the pseudo inverse of the transpose of $\hat{\mathbf{V}}_1$ and the transpose of $\hat{\mathbf{V}}_2$. An eigenvalue decomposition of $\mathbf{A}$ is performed, and the first M eigenvalues of $\mathbf{A}$ correspond to the s_k in Equation 4.13. The exponential for each sinusoidal component of the signal is derived from the real part of these eigenvalues

$$\alpha_k = \frac{\mathrm{Re}[\ln(\hat{\lambda})]}{\Delta t}. \tag{4.15}$$

For bodies with many modes of vibration, there are potentially many components to the signal. Rather than requiring one adjoint solution per component of the decomposed signal, a KS function is applied to aggregate the exponents, and the maximum growth (minimum

damping) is approximated as

$$c(\boldsymbol{\alpha}, \rho) = m + \frac{\ln\left[\sum_{k=1}^{M} e^{\rho(\alpha_k - m)}\right]}{\rho}. \tag{4.16}$$

Algorithm 7 Matrix pencil-based method for estimating and aggregating damping

Given: $\mathbf{t}$, $\mathbf{z}$, Δt, ρ
$\mathbf{Z} \leftarrow \mathbf{H}(\mathbf{t}, \mathbf{y})$ ▷ Interpolate the simulation data $\mathbf{z}$ to $N \approx \mathscr{O}(100)$ samples $\mathbf{Z}$
$L \leftarrow N/2 - 1$ ▷ Set pencil parameter L between $N/3$ and $N/2$
for $i = 1 \rightarrow N - L$ **do**
    for $j = 1 \rightarrow L + 1$ **do**
        $\mathbf{Y}_{i,j} = \mathbf{Z}_{i+j}$ ▷ Fill Hankel matrix $\mathbf{Y}$ with samples $\mathbf{Z}$
    end for
end for
$\mathbf{U}, \boldsymbol{\Sigma}, \mathbf{V}^T \leftarrow \mathrm{SVD}(\mathbf{Y})$ ▷ Singular value decomposition of Hankel matrix $\mathbf{Y}$
$M \leftarrow f(\boldsymbol{\Sigma})$ ▷ Choose model order M as a function of the singular values
$\hat{\mathbf{V}} = \mathbf{V}_{(:,J)}$ ▷ Keep only $J = 1, \ldots, M$ singular vectors
$\hat{\mathbf{V}}_1 = \hat{\mathbf{V}}_{(I,:)}$ ▷ Keep $I = 1, \ldots, L$ rows
$\hat{\mathbf{V}}_2 = \hat{\mathbf{V}}_{(I,:)}$ ▷ Keep $I = 2, \ldots, L + 1$ rows
$\mathbf{A} = \left[\hat{\mathbf{V}}_1^T\right]^+ \hat{\mathbf{V}}_2^T$ ▷ Compute $\mathbf{A}$ matrix from pseudoinverse of $\hat{\mathbf{V}}_1^T$ and $\hat{\mathbf{V}}_2^T$
$\boldsymbol{\lambda}, \boldsymbol{\Psi}, \boldsymbol{\Phi} \leftarrow \mathrm{EIG}(\mathbf{A})$ ▷ Compute eigenvalues $\boldsymbol{\lambda}$, left and right eigenvectors, $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$
$\hat{\boldsymbol{\lambda}} = \boldsymbol{\lambda}_{(I)}$ ▷ Keep $I = 1, \ldots, M$ eigenvalues
$\boldsymbol{\alpha} = \mathrm{Re}(\ln(\hat{\boldsymbol{\lambda}}))/\Delta t$ ▷ Compute damping from eigenvalues
$\hat{\alpha} = c(\boldsymbol{\alpha}, \rho)$ ▷ Use KS function c with parameter ρ to aggregate damping

The matrix pencil method has several advantages over other methods of estimating the damping. While other methods can include noise filtering as a preprocessing step, the noise term is embedded in the matrix pencil method which makes it robust. Other Prony-based methods require root finding whereas the matrix pencil method is based on eigenvalues with can be faster and more robust. Unlike the half-power and log decrement approaches, the matrix pencil method can identify growing or decaying components of a multi-component signal in an automatic and differentiable way.

4.1.3 Comparison of the Damping Prediction Methods

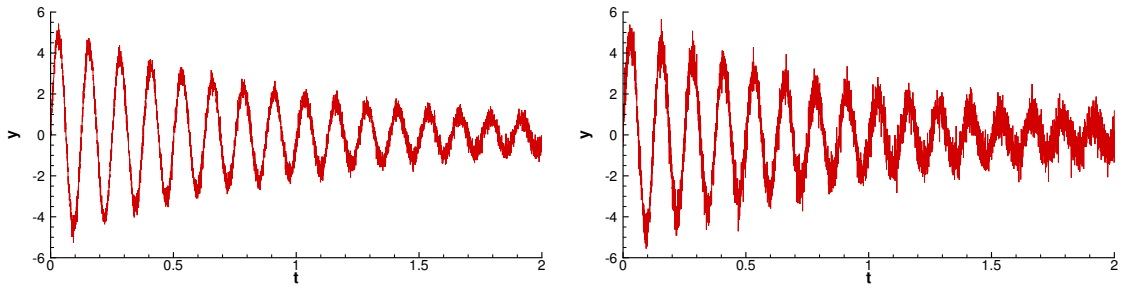
The next set of examples study the effectiveness of the different methods described above. They are studied on various signals with difficulties that maybe be encountered during a CFD analysis including noise, multicomponent signals, and inaccurate dereferencing. Finally, the methods are compared on modal responses from CFD analysis from the X-57 Maxwell [146].

Decaying Sinusoidal Signal with Noise

This study examines the tolerance of the methods to noise. The example signal is a decaying sinusoid with noise:

$$z(t) = 5e^{-t}\sin(50t) + w(t), \qquad t \in [0,2], \tag{4.17}$$

where $w(t)$ is the noise term based on a random normal distribution. The example signal for noise levels (standard deviation of the normal distribution) with 5% and 10% of the initial amplitude are given in Figure 4.4. Table 4.1 compares the predictions of the log decrement, half-power, and matrix pencil methods for levels of noise between zero and ten percent. The log decrement method is automated with a peak identification method that included height and minimum distance filtering for noise tolerance as would be done in an optimization framework. For the Hilbert transformation, the mean of the instantaneous damping is taken for the middle three-fifths of the signal to avoid the regions of high numerical error. For the half-power bandwidth method, a fast Fourier transform (FFT) converts the signal to the frequency domain. To avoid assuming a shape of the power spectrum, the half-power points are the discrete points from the FFT power spectrum closest to the half-power amplitude. The two columns of matrix pencil results are the predictions before and after the application of the KS aggregation function with $\rho = 50$. At zero noise, the log decrement, Hilbert transformation, and matrix pencil method are nearly exact (the exact exponent coefficient is



(a) 5% noise. (b) 10% noise.

Figure 4.4: Decaying sine with noise example problem.

Table 4.1: Identification of exponent coefficient (α) with the various methods for the noisy sine signal.

Noise (%)	Log Dec.	Hilbert	Half-power	Matrix Pencil	Matrix Pencil (KS)
0.0	-0.99977	-0.99769	-1.01905	-1.00000	-0.99307
0.1	-0.97772	-1.04763	-1.01905	-1.00022	-0.99329
1.0	-0.93360	-0.58932	-1.01905	-1.00142	-0.99449
5.0	-0.78456	-0.21975	-1.01905	-0.99631	-0.98938
10.0	-0.56002	-0.08667	-1.03077	-1.02429	-1.01736

-1) while the half-power method is within two percent. The error in the half-power method arises from the limited resolution of the FFT. Having a longer signal history would lead to a finer frequency resolution. As the noise level increases, the accuracy of the log decrement and Hilbert transformation method degrades. At the 10% noise level, the log-decrement-predicted exponent coefficient is in error by 44%, and the Hilbert transformation is 91.3% too low. The noise tolerance of the log decrement and Hilbert transformation methods could be improved by adding a filtering step to the process. Meanwhile without additional steps, the more noise-tolerant half-power and matrix pencil methods stay within 3.1% and 2.5% of the exact exponent, respectively, even at significant noise levels.

Sensitivity to Dereferencing

In many aeroelastic flutter problems, the equilibrium or reference point about which the oscillations occur is not $z = 0$ as was the case in the previous examples. Here dereferencing refers to the shifting of the signal to make it oscillate about $z = 0$ as required by some of

Table 4.2: Identification of exponent coefficient (α) with the various methods for improperly dereferenced signals.

Offset, z_0	Log Dec.	Hilbert	Half-power	Matrix Pencil	Matrix Pencil (KS)
0.0	-0.99977	-0.99769	-1.01905	-1.00000	-0.99307
0.01	-0.99536	-1.00805	-1.01905	-1.00000	-0.99307
-0.01	-1.00427	-0.98689	-1.01905	-1.00000	-0.99307
0.1	-0.95744	-1.10734	-1.01905	-1.00000	-0.99307
-0.1	-1.04674	-0.89515	-1.01905	-1.00000	-0.99307
0.5	-0.82181	-1.67749	-1.01905	-1.00000	-0.99307
-0.5	-1.30377	-0.53749	-1.01905	-1.00000	-0.99307

the damping identification methods. Inaccuracy in identifying the reference point can lead to errors in the damping prediction. The same signal from the first example is used except the noise term is replaced by a constant offset to model an error in the estimation of the equilibrium point:

$$z(t) = 5e^{-t}\sin(50t) + z_0, \qquad t \in [0,2]\,, \tag{4.18}$$

where z_0 is the error in dereferencing. Table 4.2 illustrates how the methods handle various errors in the reference point. The signal decomposition methods, the matrix pencil and the half-power methods, both identify components with zero frequency, i.e., the reference point, therefore the predicted damping does not change as the reference offset is varied. The log decrement and Hilbert transformation methods do show sensitivity to the dereferencing. When the offset is 10% of the initial amplitude, the log decrement method has up to 30.4% error while the Hilbert transform method has as much as 67.7% error.

Multicomponent Signal

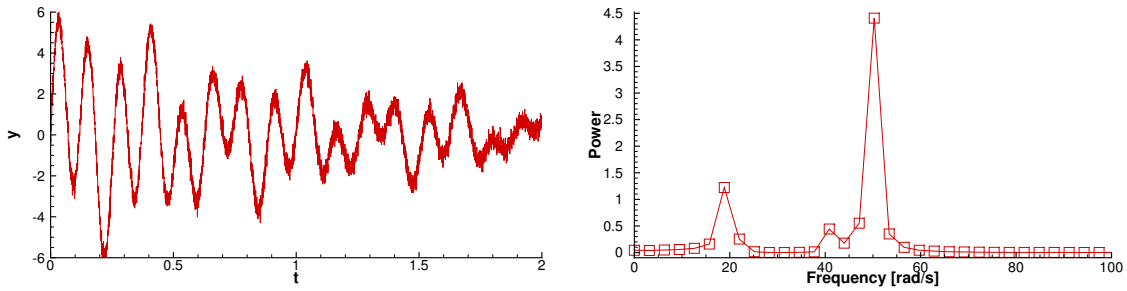
The next example is a multicomponent signal where a smaller amplitude term is growing:

$$z(t) = 5e^{-t}\sin(50t) + 2e^{-0.5t}\sin(20t) - 0.5e^{0.05t}\sin(40t) + w(t), \qquad t \in [0,2]\,. \tag{4.19}$$

The random normal noise component has a standard deviation of 0.25. Figure 4.5 illus-

trates this signal and its power spectrum. The superposition of the components of the signal makes it difficult to identify that there is an unstable mode by visual inspection. With no noticeable pattern, the log decrement method does not produce a consistent damping estimate between any sets of peaks. The relatively short length of the time signal creates a low frequency resolution for the FFT which affects the accuracy of the half-power method. The power spectrum does pick up three peaks corresponding to the three sinusoidal component of the signal. Some form of interpolation could be applied, but the approximation of the half power of the peak at 40 rad/s would be difficult since the peak is essentially a single point. Additionally, the power spectrum does not discriminate between growing and decaying components. Extra steps that reconstruct the signal with different combinations of growing and decaying components from the half-power method to find the best fit against the original signal and ultimately the correct combination of signal components can be added, but this adds complexity to the algorithm. Table 4.3 shows how the matrix pencil method decomposes the noisy multicomponent signal. The absolute value of the frequencies match the exact frequencies to three significant digits, and the addition of the amplitudes for the positive and negative frequency pairs produces amplitudes of 4.976, 1.998, 0.500 (compared to the original amplitudes of 5.0, 2.0, and 0.5). The predicted exponents are also very accurate. For the decaying pieces, the exponent error is less than 0.3%, and for the unstable piece it is 5.33%, which is very good since the standard deviation of the noise is 50% of the initial amplitude of the unstable signal. Not only is the matrix pencil method able to pick up the small unstable component of the signal, it also provides accurate exponent predictions with a relatively short time signal.

Because the Hilbert transformation was very sensitive to noise in the first example, it was applied to the same signal without the noise term. The envelope function and instantaneous damping are shown in Figure 4.6. The envelope function indicates significantly more oscillatory behavior than observed in Figure 4.2. Since the single constant frequency assumption is violated, the damping ratio prediction in Equation 4.8 is not applicable. There-



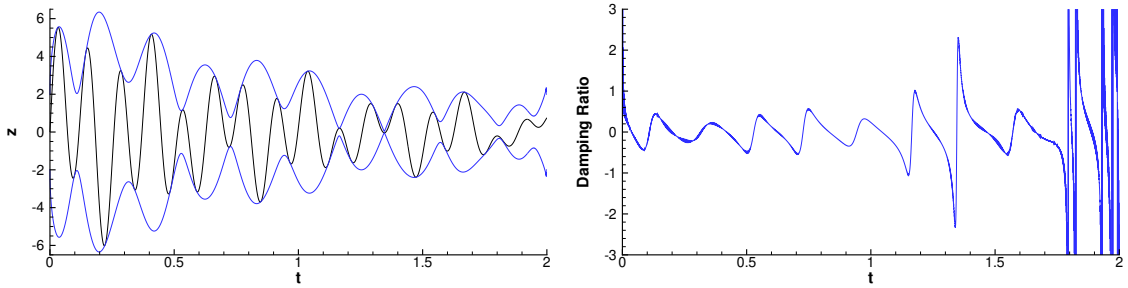
(a) Signal. (b) Power Spectrum.

Figure 4.5: Multicomponent sinusoid with noise example problem.

Table 4.3: Matrix pencil breakdown of the multicomponent sinusoid example.

Component	Frequency [rad/s]	Exponent (α_k)	Amplitude
0	50.0157709075	-0.99780444174	2.48816260236
1	-50.0157709075	-0.99780444174	2.48816260236
2	19.9956633077	-0.50051227204	0.99921844326
3	-19.9956633077	-0.50051227204	0.99921844326
4	40.0160821443	0.05266857242	0.25016878621
5	-40.0160821443	0.05266857242	0.25016878621

fore, the instantaneous damping ratio oscillates between positive and negative values and does not give a good indication of the behavior of the signal. While this is a simpler implementation of the envelope function method, it is unlikely that any envelope-function-based method would identify the unstable component of the signal because they are designed for overall behavior and the unstable component is diluted by the decaying components in the short time window.



(a) Signal and instantaneous envelope. (b) Instantaneous damping ratio.

Figure 4.6: Multicomponent signal without noise.

Matrix Pencil Sensitivity to Model Number

The noise filtering in the matrix pencil method is accomplished by selecting a model order that is lower that the full order of the Prony series decomposition problem in line 10 of Algorithm 7. If the model number is too high, the noise may not be filtered out. If the model number is too low, the matrix pencil method may not have enough components to accurately reproduce the signal. The effect of the model number is studied on the following signal

$$z(t) = 5e^{-t}\sin(50t) + 2e^{-0.5t}\sin(20t) - 0.5e^{0.05t}\sin(40t) + 0.1 + w(t), \qquad t \in [0,2], \tag{4.20}$$

where the noise level $w(t)$ has a standard deviation of 0.5 which is 10% of the largest amplitude component. Ideally, the matrix pencil method would select seven as the model number that accounts for the constant offset and a positive and negative frequency term for the three sinusoidal components in the signal. Table 4.4 shows how the matrix pencil method's decomposition of signal changes as the model number is increased. With a model number of two, the matrix pencil pencil method approximates the signal with a damped sinusoid that is close to largest component of the signal. As the model number increases to seven, components zero and one become closer representations of the original signal's largest component, and the other components of the signal begin to appear. When the model number is six, the smallest amplitude oscillation appears in the decomposition. For this signal it is important to capture this new component because it is the harmonic that is undergoing exponential growth. At a model number of seven, the decomposition is most accurate, as expected. The least accurate value at a model number of seven is the exponent coefficient of the growing signal with an absolute and relative error of 0.0028 and 5.6% respectively. When the model number is above seven, some of the noise is retained as seen in the high frequency components in the final two columns of the table. For a model number of eight, the last component is a high-frequency, low-amplitude oscillation which has an

extremely large damping that would not affect the ability of the KS function aggregation to determine the maximum exponent coefficient. As the model number is increased to nine, the noise contributions appear as unstable high frequency components. The noise components dominate the KS aggregation of the exponent coefficients, which is 0.283. This would be a undesirable situation during an optimization because the optimizer would try to reduce the damping of a signal component that is not physically relevant. From this analysis, there are only two choices of model number, six and seven, that include enough components to capture the important oscillatory terms but do not retain the noise.

In general, the best model number will not be known *a priori*. One approach to select the model number is based the magnitude of the singular values. Figure 4.7 shows the singular values normalized by the largest value for noise levels of 1% and 10%. The first six singular values appear as pairs of approximately equal value. The seventh value is one order of magnitude smaller than the largest singular value. These first seven singular values are essentially equal for the two different noise levels. The remaining singular values are related to the noise in the signal. The one order of magnitude difference of the noise-related singular values observed between the noise levels is consistent with the difference in noise level. Because the noise term is random, it is logical that the energy would be approximately equally distributed among the range of the possible frequencies, leading to approximately equal singular values. One distinguishing feature that delineates the true signal singular values from the noise is the significant drop of magnitude between the lowest true component magnitude and those of the noise. Therefore, the model number in optimizations presented in this work is calculated as the number of normalized singular values that are above the largest singular values times a tolerance. For the optimizations presented in this work, this tolerance is set between 0.1 and 0.01 under the assumption that RANS-based simulations will not have the level of noise of this example signal.

Table 4.4: Matrix pencil method's sensitivity to the model number parameter.

	Model #	2	3	4	5	6	7	8	9
Component 0	Frequency	49.7461	49.7848	49.8486	49.8539	49.9990	49.9978	50.0069	50.0050
	Amplitude	2.4144	2.3739	2.4943	2.5102	2.5024	2.5040	2.4809	2.4595
	α	-0.9480	-0.9164	-1.0035	-1.0202	1.0103	-1.0019	-0.9894	-0.9723
Component 1	Frequency	-49.7461	-49.7848	-49.8486	-49.8539	-49.9990	-49.9978	-50.0069	-50.0050
	Amplitude	2.4144	2.3739	2.4943	2.5102	2.5024	2.5040	2.4809	2.4595
	α	-0.9480	-0.9164	-1.0035	-1.0202	1.0103	-1.0019	-0.9894	-0.9723
Component 2	Frequency		0.0000	20.0375	20.0622	19.9872	19.9987	20.0072	20.0229
	Amplitude		0.2199	0.9899	1.0172	1.0103	1.0001	0.9970	1.0064
	α		-	-0.4726	-0.5223	-0.5042	-0.5000	-0.4912	-0.5006
Component 3	Frequency			-20.0375	-20.0622	-19.9872	-19.9987	-20.0072	-20.0229
	Amplitude			0.9898	1.0172	1.0103	1.0001	0.9970	1.0064
	α			-0.4726	-0.5223	-0.5042	-0.5000	-0.4912	-0.5006
Component 4	Frequency				0.0000	39.9903	40.0040	40.0257	40.0735
	Amplitude				0.1443	0.2429	0.2491	0.2364	0.2543
	α				-	0.0729	0.0528	0.0900	0.0400
Component 5	Frequency					-39.9903	-40.0040	-40.0257	-40.0735
	Amplitude					0.2429	0.2491	0.2364	0.2543
	α					0.0729	0.0528	0.0900	0.0400
Component 6	Frequency						0.0000	0.0000	0.0000
	Amplitude						0.0989	0.1262	0.1011
	α						-	-	-
Component 7	Frequency							7852.41	2736.23
	Amplitude							0.0787	0.2766
	α							-1788.46	0.0125
Component 8	Frequency								-2736.23
	Amplitude								0.2766
	α								0.0125

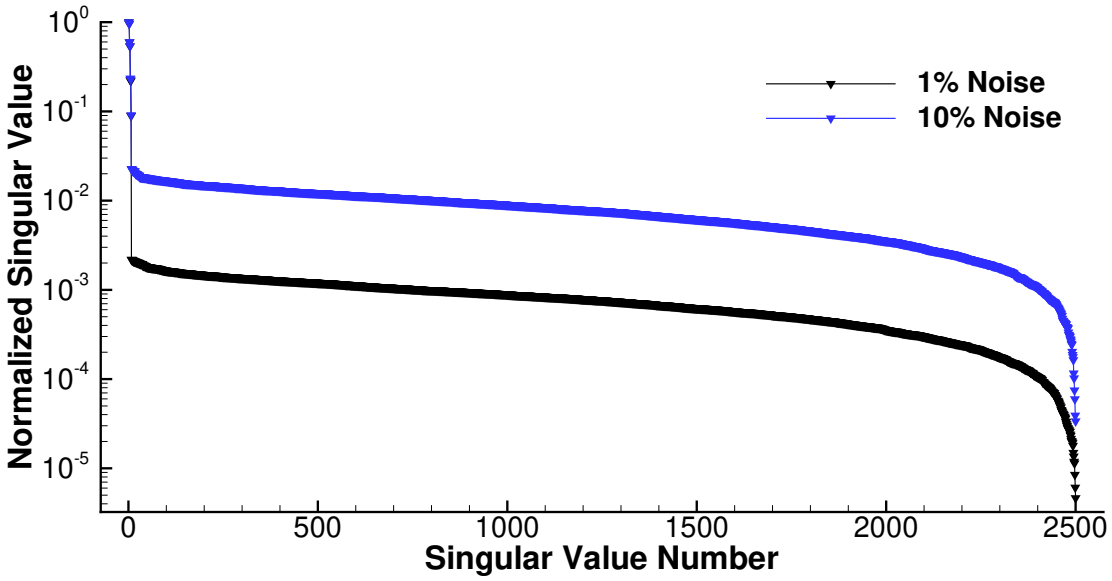
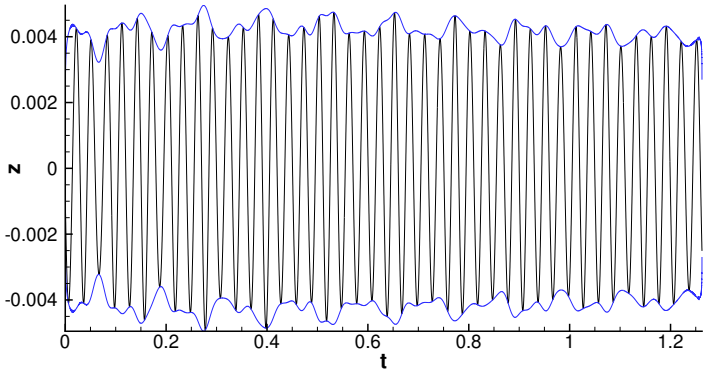


Figure 4.7: Normalized singular values for the decomposition of Equation 4.20.

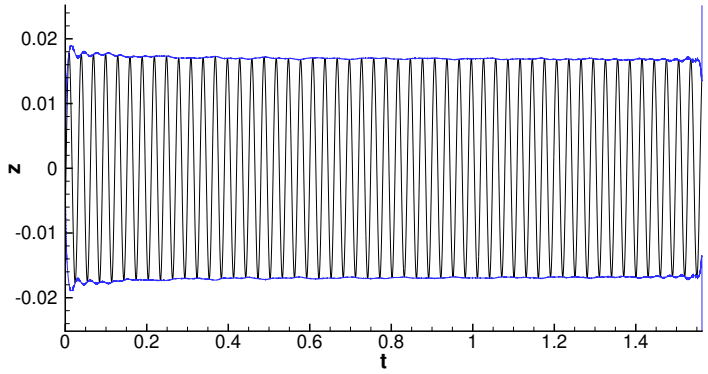
X-57 Maxwell System Identification

The final comparison between the damping calculation methods is with modal responses of a structure coupled to CFD. While the other tests have allowed verifying methods against exact results and testing sensitivity to particular signal features, this test represents more realistic data, but the true signal decomposition is not known. For this type of problem, consistent results across the methods is the best indication that they are working well. The modal displacements are taken from the X-57 Maxwell simulations with FUN3D from Heeg et al. [146]. Four modes (Figure 4.8) are selected as representative of the types of behaviors observed in the full set of modes. As would be done during optimization, an initial time period has been removed so that the initial transients after the modal excitations have disappeared and do not affect the calculated damping. Mode six is a relatively clean signal with very low damping. Mode five is a very low damping response with more variation in amplitude. Mode seven represents a sinusoid with higher damping, and mode nine is a decaying signal with multiple components.

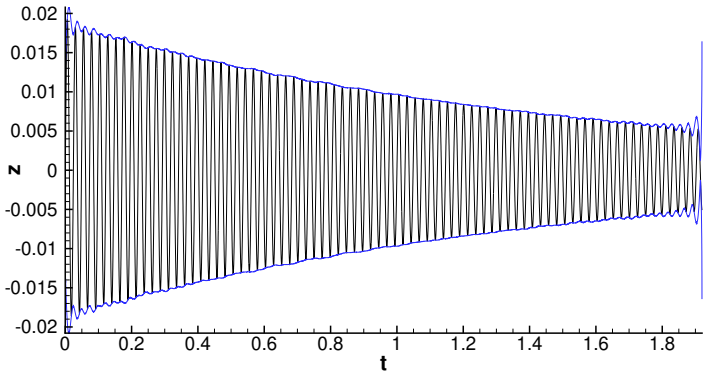
Table 4.5 gives the predicted damping ratios for different modes with the various system identification methods. For first three modes, the table illustrates that there is reasonably



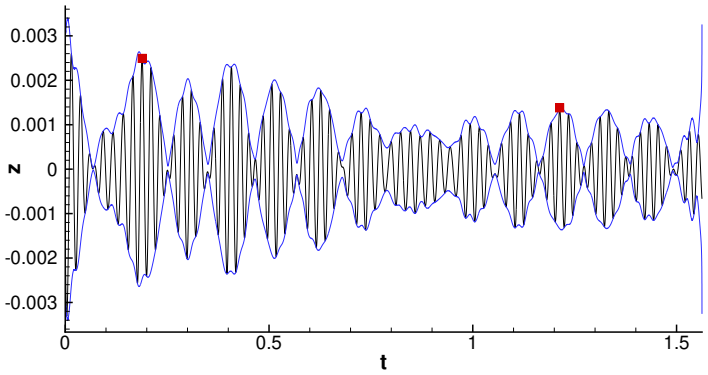
(a) Mode 5.



(b) Mode 6.



(c) Mode 7.



(d) Mode 9.

Figure 4.8: X-57 modal responses with instantaneous envelope functions.

good agreement between Hilbert transform and matrix pencil methods. Even for the very low damping of responses of modes five and six, the Hilbert transformation and matrix pencil damping ratios different by less than 2.0×10^{-5}. For mode seven, they agree to within 1% of each other. The log decrement method also agrees fairly well with the Hilbert transformation and matrix pencil methods for the first modes, but predicts slightly higher damping for the modes near zero damping. For mode nine, the log decrement method using the peaks indicated by the red marks in Figure 4.8d is 2.38 times larger than than of the matrix pencil method. The Hilbert transformation method's damping ratio is about two orders of magnitude larger than the matrix pencil and log decrement results. Since Equation 4.8 in the Hilbert method assumes a single mode while the log decrement and matrix pencil methods are relatively close, it is likely the Hilbert method is less accurate than the other two approaches. The damping ratios predicted by the half-power bandwidth method are listed as plus or minus because the method does not indicate whether a component of the signal is growing or decaying. The half-power bandwidth method does not perform well for determination of damping of these signals. The lengths of the signals create FFT frequency resolutions of approximately 5 rad/s. Linear interpolation of the FFT-based power spectrum approximates the half-power frequencies. The limited resolution causes the calculated damping of mode seven to be lower than that of modes five and six, which is incorrect. While the minimum damping is close to the matrix pencil KS aggregated value for mode nine, this appears to be coincidental as the half-power bandwidth and matrix pencil methods predict that the minimum damping ratio occurs on different modes. Table 4.6 lists the predicted frequencies computed from the matrix pencil and half-power bandwidth methods. While these methods indicate the minimum damping occurs on different modes, the frequencies are within the resolution of the FFT frequencies (4.02 rad/s). That is a good indication that the signal decompositions for the half-power bandwidth and matrix pencil methods would be similar if the FFT resolution was refined.

Table 4.5: Predicted damping ratios of X-57 modal responses.

Mode Number	Log Decrement	Hilbert	Half-power (min)	Matrix Pencil (KS)
5	0.00033542	0.00014745	$\pm$ 0.00170843	0.00015013
6	0.00017212	0.00008456	$\pm$ 0.00185588	0.00007053
7	0.00264498	0.00258143	$\pm$ 0.00089262	0.00255767
9	0.00203437	0.22148883	$\pm$ 0.00092640	0.00085409

Table 4.6: Frequencies (rad/s) predicted for X-57 mode nine response. The starred frequencies indicate the mode with the smallest predicted damping

Component	Half-power	Matrix Pencil
1	257.4*	257.6
2	289.5	288.2
3	313.7	312.0*
4	317.8	319.4

Summary of Damping Prediction Method Comparisons

Despite its sensitivity to the model number, the matrix pencil method appears to be the best suited method for identifying damping coefficients in CFD-based flutter analysis and optimization. The log decrement method is not robust enough to consistently handle multicomponent signals. The Hilbert transformation-based envelope function of Zhang [89] does not handle multicomponent signals or noise in an effective manner. While able to decompose the signal into different components, the half-power bandwidth method's resolution is insufficient for time windows that would be typical of CFD calculations. Curve fits other than linear interpolation could be applied to the low resolution power spectrum, but any assumed shape of the power spectrum will heavily influence the damping prediction. The matrix pencil method is noise tolerant and capable of handling multicomponent signals with growing and decaying components. Therefore, the matrix pencil method has been selected as the basis of flutter-based optimization in this work. The following sections demonstrate its application in flutter-based optimizations.

Table 4.7: NACA 64A010 structural properties for flutter identification.

Property	Value
Semichord, b [m]	0.5
Elastic axis location, x_{EA}/b (measured from midchord)	-0.45
Center of mass location, x_c/b (measured from elastic axis)	0.3
Mass ratio, $\mu = m/\pi\rho_\infty b^2$	20
Radius of gyration, r/b	0.5
Plunge natural frequency [rad/s]	100.0
Pitch natural frequency [rad/s]	100.0

4.1.4 NACA 64A010

The matrix pencil method is next applied in the context of adjoint-based optimization for flutter identification of a pitching and plunging NACA 64A010 airfoil. For this flutter identification process described in Section 4.1.1, the object is to minimize the dynamic pressure subject to a constraint that the KS aggregated damping from the matrix pencil method is zero. The dynamic pressure is also the only design variable. This flutter identification process was applied between Mach 0.3 and 0.7 in increments of 0.1.

The structural model in TACS is formulated in two degrees of freedom with a rotational (pitch) and translational (plunge) spring attached at the elastic axis of the airfoil. The properties of the structure are provided in Table 4.7. The pitch and plunge motions are governed by the following equations

$$m\ddot{h} + S_\alpha \ddot{\alpha} + m\omega_h^2 h = -L, \tag{4.21}$$

$$S_\alpha \ddot{h} + I_\alpha \ddot{\alpha} + I_\alpha \omega_\alpha^2 \alpha = M_y, \tag{4.22}$$

where m is the mass per unit span, S_α is the static unbalance, I_α is the sectional moment of inertia about the elastic axis, h is the plunge, α is the angle of attack or pitch in radians, L is the sectional lift, and M_y is the sectional moment about the elastic axis. These equations were integrated numerically with a second-order BDF scheme.

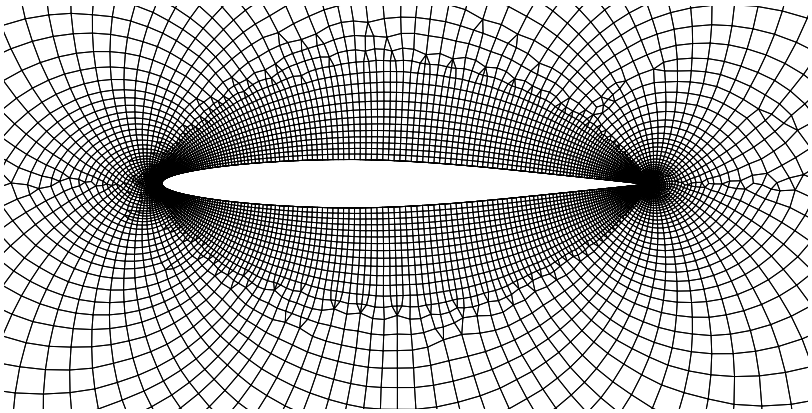
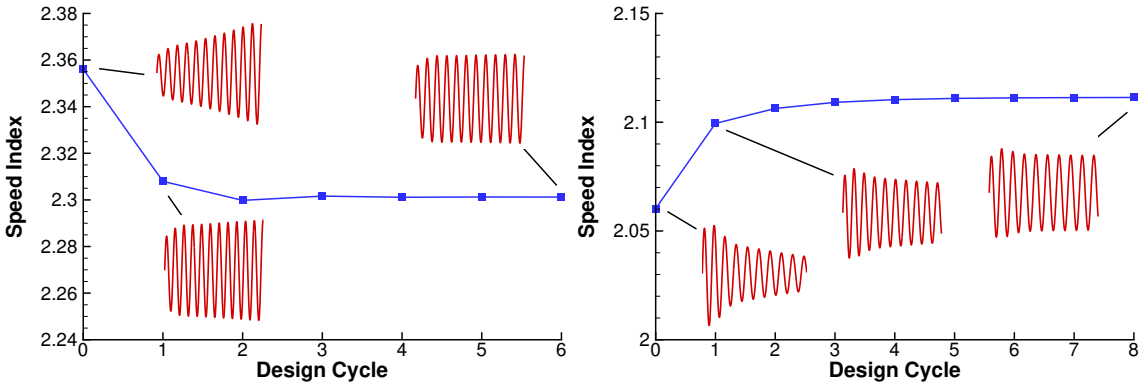


Figure 4.9: Aerodynamic mesh near the NACA 64A010.

The aerodynamics were modeled as an inviscid compressible flow within FUN3D. Since the adjoint implementation requires that FUN3D be run on three-dimensional problems, the aerodynamic mesh shown in Figure 4.9 contains three identical but offset planes, each with 11,703 nodes. The BDF2opt integration scheme in FUN3D was applied to integrate the aerodynamic governing equations. The time step sizes in the simulations were chosen so that there were at least 40 time steps per oscillation cycle. Five subiterations per time step were used, which typically led to about five orders of magnitude of convergence during the pseudo-time stepping. The motion type in FUN3D was set to 'rigid+deform'.

The simulations were performed for approximately 10 oscillation cycles (between 4000 and 8000 time steps depending on the Mach number). This number of cycles is long enough for the initial transients to disappear and for the matrix pencil method to consistently calculate the damping. Since this is a coupled two degree-of-freedom flutter problem, the matrix pencil method was only applied to the pitch signal. Only the second half of this signal was fed to the damping calculation to avoid the initial transients affecting the results.

The sequential least-squares quadratic programming (SLSQP) optimizer [147] in pyOpt [148] was utilized to find the flutter point. Each optimization problem converged in six to eight design cycles. Figure 4.10 illustrates the optimization convergence to the flutter speed for Mach numbers of 0.3 and 0.5. The speed index is a nondimensional number that defined as the ratio of the free-stream velocity to the product of the semichord of the airfoil



(a) Mach 0.3.

(b) Mach 0.5.

Figure 4.10: NACA 64A010 flutter point identification optimization histories. The red lines illustrate the pitch response at points during the optimization.

and the natural frequency of the pitching motion

$$V_s = \frac{V_\infty}{\omega_\alpha b}. \tag{4.23}$$

For the results at Mach 0.3, the initial dynamic pressure produces unstable oscillations, so the optimizer decreased the dynamic pressure. As the dynamic pressure is decreased, the rate of oscillation magnitude growth decreases, which is evident in the pitch response at the first design cycle. After six design cycles, the oscillations experience neither growth nor decay indicating that the flutter point has been found. For the results at Mach 0.5, the initial response is a damped oscillation. To satisfy the damping equality constraint, the optimizer increased the dynamic pressure. At the eighth design cycle, there is some initial growth and decay of the response, but the latter half of the response where the matrix pencil method is applied has essentially zero damping.

Figure 4.11 relates the predicted flutter speed index as a function of Mach number. The red curve is from linear Theodorsen theory [149] with a Prandtl-Glauert compressibility correction [150] applied to the loads. The differences is likely due to the finite thickness of the airfoil in the simulation whereas the Theodorsen theory is based on a thin airfoil assumption. There is little change in the flutter speed index between the Mach numbers of

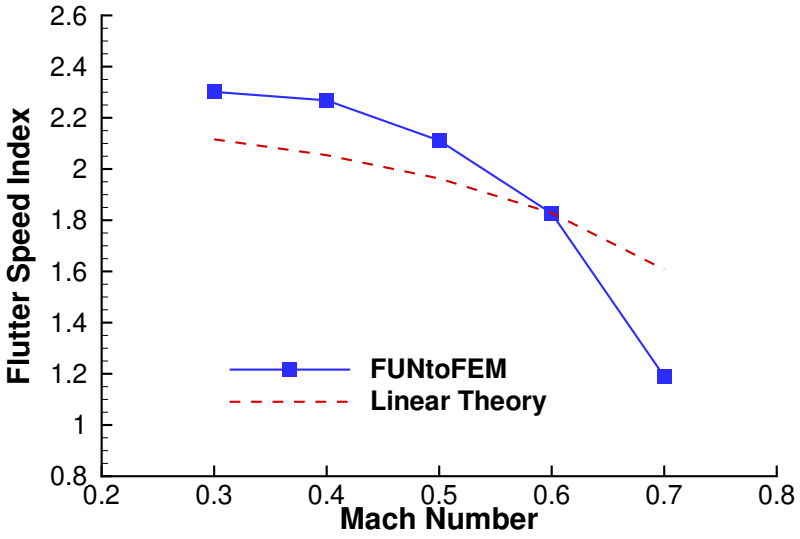


Figure 4.11: NACA 64A010 flutter boundary.

0.3 and 0.4. As the Mach number is further increased, the compressibility effects become more prominent, and the flutter speed drops significantly. At Mach 0.7, the flutter speed index is 52% of the value at Mach 0.3. This example illustrates that the matrix pencil method can be applied to signals generated by CFD and works within an optimization framework.

4.1.5 AGARD 445.6 Wing

Next, the matrix pencil method is applied to a three-dimensional flutter problem, the AGARD 445.6 wing. Experimental flutter data from the Transonic Dynamics Tunnel at NASA Langley was published by Yates [151]. Since then, the AGARD 445.6 wing has widely served as a flutter case for benchmarking computational aeroelasticity tools. The wall-mounted wing has a span of 2.5 feet, root chord of 1.83 feet, a quarter chord sweep angle of 45 degrees, and a taper ratio of 0.66. The airfoil is a symmetric NACA 65A004 airfoil, and all flutter data for the test case studied in this work is at zero degrees angle of attack.

The structural model described by Yates is a modal structure; therefore, an adjoint-enabled modal solver was written and coupled to FUNtoFEM for this problem. The first four modes of the structure were retained for the analysis. The aerodynamics were mod-

eled with a 446,584 node mesh and Euler simulation in FUN3D. Each flutter simulation consisted of 6,000 time steps with 125 steps before the excitation of the modes in order to approximately reach a steady state.

The optimizations performed were a similar set up to the NACA 64A010 flutter identification optimizations. The only design variable is dynamic pressure, and the objective is to minimize the dynamic pressure subject to a constraint that the KS aggregated damping predicted by the matrix pencil method is equal to zero. The matrix pencil method was applied to the final 3,000 time steps of each simulation. This process was performed for Mach numbers in the range of 0.499 to 1.141.

Figure 4.12 compares the flutter dynamic pressure determined by the FUNtoFEM optimization to the experimental value as well as some FUN3D modal solver results from Silva et al. [152]. The optimizations based on the matrix pencil method successfully determined the flutter dynamic pressures to within 0.1 psf across the range of Mach numbers in four to seven design cycles. Like the inviscid FUN3D modal solution from Silva et al., the FUNtoFEM solution predicts a transonic flutter dip at M_∞=0.96 but overpredicts the flutter dynamic pressure above Mach 1. The discrepancy between the FUNtoFEM and inviscid FUN3D results are likely due to the difference in mesh resolution. The mesh in the inviscid FUN3D results is two million nodes, while the FUNtoFEM mesh has less than five hundred thousand nodes. Since the purpose of this study is to explore the effectiveness of the matrix pencil based flutter constraint rather than achieve spatially converged solutions, the discrepancy between the FUN3D and FUNtoFEM Euler solutions was deemed acceptable. The capturing of the correct trends implies that the fidelity is sufficient to model the important physics in the problem.

Figure 4.13 shows the convergence of the flutter identification optimization for $M_\infty = 1.072$ from various initial dynamic pressures. The time histories for the first mode (the flutter mode) at some points is also given in the plot to illustrate the convergence from certain initial dynamic pressures. The highest initial dynamic pressure is well above the flutter

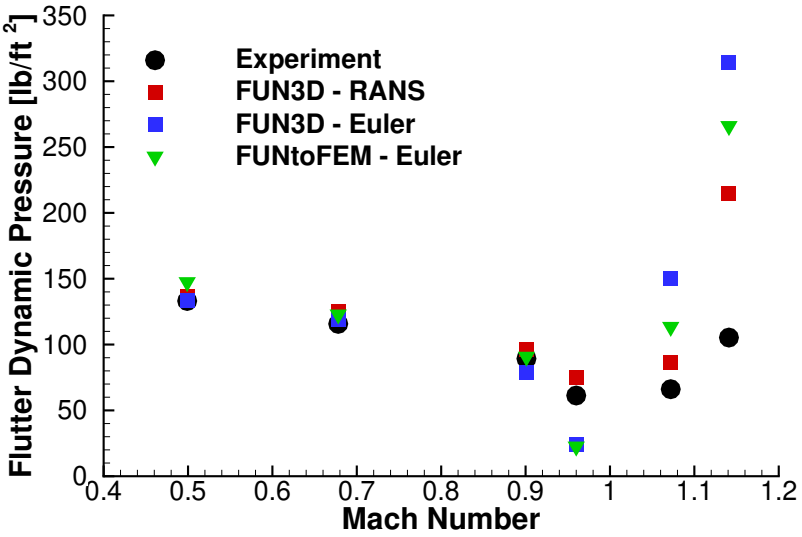


Figure 4.12: AGARD 445.6 flutter boundary. FUN3D results are from Silva et al. [152].

dynamic pressure, so the increasing magnitude of the oscillations causes the mesh deformation to produce negative cell volumes. This causes the simulation and optimization to fail. From the initial dynamic pressures of 144.0 psf and 100.8 psf, the optimizations converge to the same dynamic pressure. At the final design cycles, the damping ratio for both of the these optimizations is less than 1.0×10^{-5}, indicating that it is the flutter dynamic pressure. Althought the initial dynamic pressures are close, the optimizations that start from $q = 43.2$ psf and 57.6 psf diverge because they start on either side of the maximum damping peak. At $q = 43.2$ psf both the damping and the derivative of the damping with respect to the dynamic pressure are positive which forces the optimization toward the $q = 0$ local minimum. Because the damping coefficient at $q = 57.6$ psf is 0.72 and near to the maximum damping, the derivative of the damping coefficient with respect to the dynamic pressure is -0.005972 which has a negative but very shallow slope. Therefore, the optimizer takes a large step to 178.15 psf which is where the tangent line intercepts zero damping. At $q = 178.15$, the simulation again fails due to negative volumes generated in the CFD volume mesh caused by unbounded oscillations. Since only two out of five optimizations converged to the desired local minimum, the success of a flutter identification optimization is observed to be sensitive to the initial conditions. In the case of the AGARD wing, an initial condition can

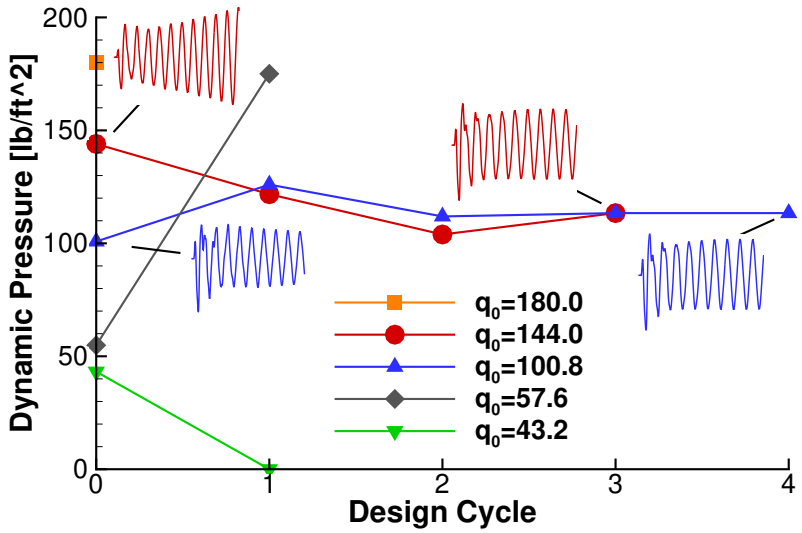


Figure 4.13: AGARD 445.6 flutter optimization history at M_∞=1.072 from various initial conditions.

be selected intelligently because there are existing experimental and computational results. However, this sensitivity is a potential robustness issue for optimization.

4.1.6 Robustness and Computational Cost of Time-accurate Flutter Constraints

In this section, a methodology for time-accurate flutter constraints has been presented based on the matrix pencil method. The matrix pencil method proved to be the most robust method for identification of fluttering modes of a system. While the matrix pencil method is sufficient for identifying damping in time domain signals, gradient-based optimization with this method is not robust for flutter identification. There are multiple conditions that give zero damping but only a subset of those conditions correspond to the desired flutter point. This makes the success of flutter identification sensitive to initial conditions. Even if the initial condition is selected to be near the initial flutter point, a large change in design variables between design cycles could cause the dynamic pressure in a flutter identification constraint to leave the neighborhood of the flutter dynamic pressure and force a situation similar to those in Figure 4.13 where the dynamic pressure converges towards a value other than the flutter point (e.g. zero dynamic pressure) or the simulation fails due to overly

large of deflections. In certain optimization algorithms, the production of negative volumes due to a too high dynamic pressure is recoverable because a failing analysis can signal the optimizer to reduce the size of the step in the design variables, but convergence to wrong minima is relevant to all local (gradient-based) optimization methods.

Time-accurate CFD for flutter clearance constraints is also currently not practical for full aircraft aeroelastic design. The large number of time-accurate CFD analyses required to sufficiently cover the flight envelope is not feasible even if a flutter clearance constraint is utilized instead of a flutter identification constraint. One potential strategy to reduce the computational cost is to select a subset of the points from the flight envelope that are thought to be the most critical. These points would be included as flutter clearance constraints in an aircraft optimization. Once the optimization converges, the flutter properties of the remainder of the flight envelope would be analyzed. If any points are above the flutter clearance damping tolerance, they can be added to the optimization as constraints, and the optimization would be restarted. This process could be repeated until the damping across the flight envelope is sufficient.

An alternative to reducing the cost of the time-accurate constraints is to limit the scope of the time-accurate CFD to flight conditions where it is necessary. While time-accurate CFD is applicable to a wide range of problems, methods such as frequency domain solvers can be just as accurate and one to two orders of magnitude faster for flutter analysis. These methods are also better suited for flutter analysis because they can produce eigenvalue data from which the damping can be directly extracted rather than time-accurate analysis which requires system identification techniques to infer the damping characteristics. While this work has shown that flutter-constrained design with time-accurate CFD is possible, there are potentially more efficient alternatives for flutter-based constraints, and time-accurate CFD will be better suited for nonlinear aperiodic problems or other nonlinear effects such as blowing or suction.

4.2 Gust Response Constraints

Gust response is another critical aeroelastic constraint on aircraft design. In the Federal Aviation Administration's (FAA) advisory circular (AC) for gust loads [153], they outline acceptable methods for modeling an aircraft experiencing a gust. In the AC, they state that "nonlinear solution methods are necessary for airplane and flight control systems that are not reasonably or conservatively represented by linear analysis models." As aircraft become more flexible, the nonlinear effects will dominate as larger deflections are encountered during gusts.

The FAA defines two types of gusts that an aircraft must be able to withstand: a discrete gust which represents a single extreme turbulence event, and a continuous gust which represents a longer duration turbulence encounter. Because the discrete gust is shorter duration, this work will focus on the discrete gust although a continuous gust could be approximated by the superposition of sine and cosine gust components in FUN3D. The discrete gust model in the AC is one-dimensional with the gust velocity normal to the flight direction. The gust is spatially uniform in the planes perpendicular to the direction of travel and has a 1-cosine velocity profile:

$$U(x) = \frac{U_{ds}}{2}\left(1 - \cos\left(\frac{\pi x}{H}\right)\right), \tag{4.24}$$

where H is the gust gradient distance or the distance to the maximum gust velocity, and U_{ds} is the maximum gust velocity. The FAA guideline says that the gust gradient distance should range from 30 to 350 feet, and the maximum gust velocity is determined from the gust gradient distance, the reference airspeed, and the altitude of the aircraft:

$$U_{ds} = U_{ref} F_g \left(\frac{H}{350}\right)^{1/6}, \tag{4.25}$$

where U_{ref} is a measure of the turbulence intensity and is given in Federal Aviation Regu-

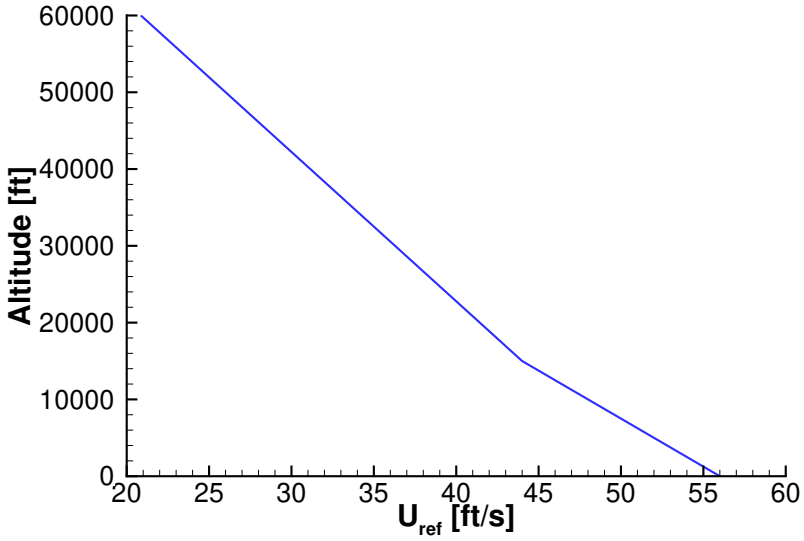


Figure 4.14: Variation of the discrete gust reference velocity, U_{ref}, with altitude.

lations (FAR) Part 25.341. In Figure 4.14, U_{ref} varies linearly between three points: 56.0 ft/s at sea level, 44.0 ft/s at 15,000 ft, and 20.86 ft/s at 60,000 ft. F_g is the flight profile alleviation factor which is defined at sea level as

$$F_{zg} = 1 - \frac{h_{max}}{250000} \tag{4.26}$$

$$R_1 = \frac{Maximum\ Landing\ Weight}{Maximum\ Takeoff\ Weight} \tag{4.27}$$

$$R_2 = \frac{Maximum\ Zero\ Fuel\ Weight}{Maximum\ Takeoff\ Weight} \tag{4.28}$$

$$F_{mg} = \sqrt{R_2 \tan\left(\frac{\pi R_1}{4}\right)} \tag{4.29}$$

$$F_g = 0.5(F_{gz} + F_{gm}) \tag{4.30}$$

From sea level, the flight profile alleviation factor linearly increases to 1.0 at the certified maximum altitude, h_{max}. There are additional considerations for aspects such as wing-mounted engines and stability, but as a first step into CFD-based gust-constrained optimization, the analysis will be limited to the general 1-cosine discrete gust.

Table 4.8: Lift sensitivity verification for a coupled CRM simulation with a gust.

	Structural Thickness	Angle of Attack	Shape Variable
Lift Adjoint	-2.60701434865e-05	0.00456874796182	-0.000711916380**269**
Lift Complex	-2.60701434866e-05	0.00456874796182	-0.000711916380**507**

4.2.1 Field Velocity Method of Gust Modeling

There are various ways to model gusts. One such method is the field velocity method [154] where the gust is created in the flow field by modifying the grid velocity term for all the nodes (except the surface nodes) without actually changing the grid displacements. Actual forced motion of the grid would also create non-physical inertial forces in the structure. The new CFD grid velocities are found by subtracting the gust velocity from actual grid velocity associated with the grid motion

$$\dot{x}\hat{i} + \dot{y}\hat{j} + \dot{z}\hat{k} = (\dot{x}_0 - u_g)\,\hat{i} + (\dot{y}_0 - v_g)\,\hat{j} + (\dot{z}_0 - w_g)\,\hat{k}. \tag{4.31}$$

The change in grid speed affects the fluxes in the governing equations of the flow and essentially acts as a source term to generate the gusts. Bartels added a field velocity method for gust modeling in FUN3D [155] which will be the basis of this work. In FUN3D the gust convects with the free-stream velocity. The gust are uniform in the planes normal to the direction of propagation. The profile of the discrete gust can be generated as a superposition of sine, cosine, 1-cosine, and Gaussian profiles. The length; start time; start position; and x, y, and z magnitude of each gust component can be set independently to form complex gust profiles.

In Reference [155], Bartels verified the field velocity method against other published results. For this dissertation, the FUN3D's adjoint was modified to allow adjoint-based gradients when the field velocity method is used. In Table 4.8, the adjoint-based sensitivities agree to at least nine significant digits with the complex step method for a RANS-based gust simulation of the CRM.

4.2.2 NACA 64A010 Gust Optimization

To demonstrate that the gust constraint and sensitivity analysis works in an optimization setting, an optimization was performed with a NACA 64A010 airfoil. The symmetric airfoil at zero degrees angle of attack is attached to a spring and allowed to translate vertically. During the simulation, there is an initial period where the steady flow field is established, and then airfoil encounters a 1-cosine vertical gust as defined in the FAA advisory circular. The optimization problem is to minimize the spring stiffness subject to the constraint that the maximum displacement of the airfoil as approximated by a KS aggregation function remain below an upper limit of 0.1 ft:

$$\begin{aligned} \min \quad & k \\ \text{subject to:} \quad & KS(z(t)) - 0.1 \leq 0. \end{aligned} \tag{4.32}$$

where the KS function is the aggregation over the time history of the simulation

$$KS(z(t)) = m + \frac{1}{\rho} \log \int_0^T e^{\rho(z(t)-m)} dt, \tag{4.33}$$

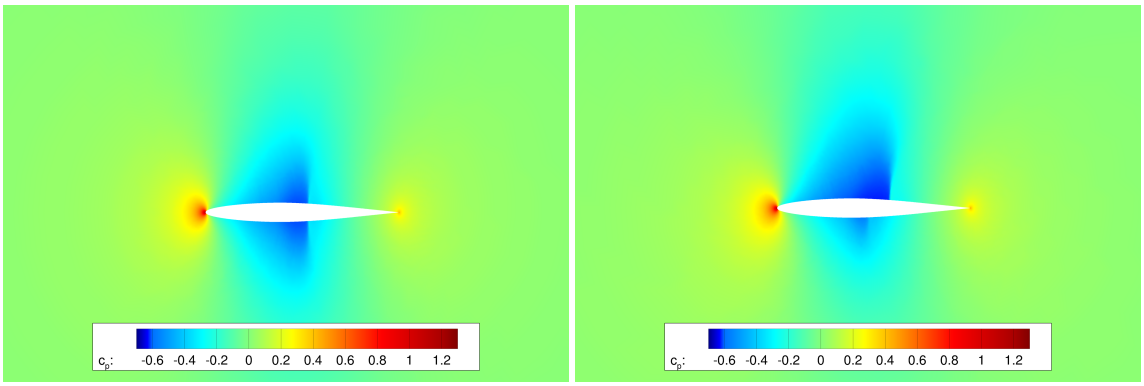
where m is the maximum displacement and ρ is a constant set to 5000.0 for this optimization. This optimization is analogous to a mass minimization subject to a stress constraint. A higher stiffness spring would imply more mass and the stress would be related to the displacement of the airfoil.

The assumed conditions are listed in Table 4.9. The gust encountered is the sharpest gust that designs are expected to withstand as defined by the FAA. Given the properties in Table 4.9, the maximum gust velocity is Mach 0.022. With the initial spring stiffness, the maximum displacement of the airfoil is 0.36 ft which is 3.6 times higher than the acceptable displacement.

For the computational model in FUN3D, the same inviscid mesh from the flutter identifications in Figure 4.9 was utilized. The selected time step size was determined so that

Table 4.9: Assumed properties and conditions for the NACA 64A010 gust optimization.

Property	Value
Altitude [ft]	30,000
Free-stream Mach number	0.8
Flight profile alleviation factor, F_g	0.92
Gust gradient distance, H [ft]	30.0
Airfoil chord [ft]	1.0
Airfoil mass per span [slug/ft]	0.05
Initial spring stiffness [lb/ft]	200.0



(a) Steady state.

(b) Peak gust amplitude at leading edge.

Figure 4.15: Pressure coefficient contours from the NACA 64A010 gust optimization. The critical pressure coefficient is -0.435.

there are 160 steps within the duration of the gust duration. Ten subiterations per time step produced at least three orders of magnitude of convergence per time step. There are 1000 total time steps with approximately 50 steps before the gust is encountered and about five times the gust duration simulated after the gust passes. The motion type in FUN3D was set to 'rigid+deform'. Figure 4.15a shows the steady state flow field at Mach 0.8. The flow is transonic with shocks on the upper and lower surface of the symmetric airfoil at $x/c = 0.53$. When the gust hits the airfoil in Figure 4.15b, the upper surface shock becomes stronger and shifted aft to $x/c = 0.58$ while the lower shock weakens at shifts to $x/c = 0.47$.

The optimization was performed with the SLSQP optimizer. Each forward and adjoint analysis took one and two hours of walltime, respectively, on 48 compute cores. The optimization history is depicted in Figure 4.16. The optimization converged with ten total

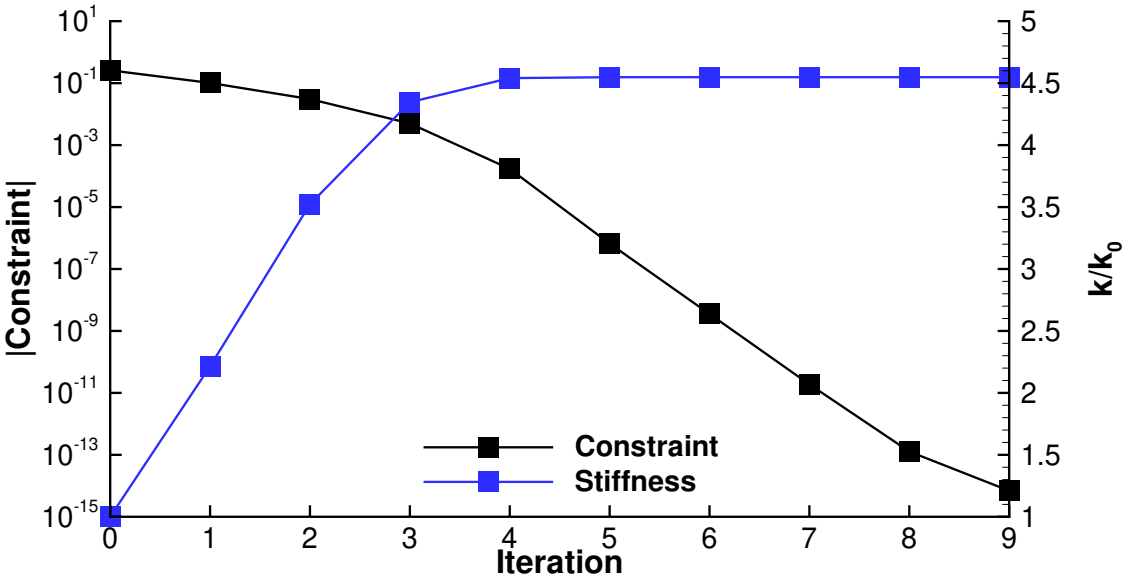


Figure 4.16: Convergence of the NACA 64A010 gust optimization.

forward analyses and nine adjoint evaluations. The stiffness of the spring increased by a factor of 4.55 over the course of the optimization. The final constraint value is 6.9e-15 indicating that the KS aggregated maximum displacement is 0.1 to machine precision.

Figure 4.17 shows the displacement histories of the initial and optimal designs. While the aggregated maximum displacement is 0.1 to machine precision, the true maximum displacement for the optimized design is 0.1055; therefore, the optimal design's displacement goes slightly above the upper bound of 0.1 in the plot. The difference between the aggregated and true maximum displacement could be reduced by increasing the value of ρ in the KS function in Equation 4.33.

Even with the simple structural model, this scenario represents a nonlinear aeroelastic problem. The presence of the shocks and their movement means that this case would be extremely difficult to capture with a lower fidelity aerodynamics model. With this simplified optimization establishing that the aeroelastic gust analysis and sensitivities can be applied successfully to complex aeroelastic physics, a gust-based constraint will be applied to a more realistic structure in the following chapter.

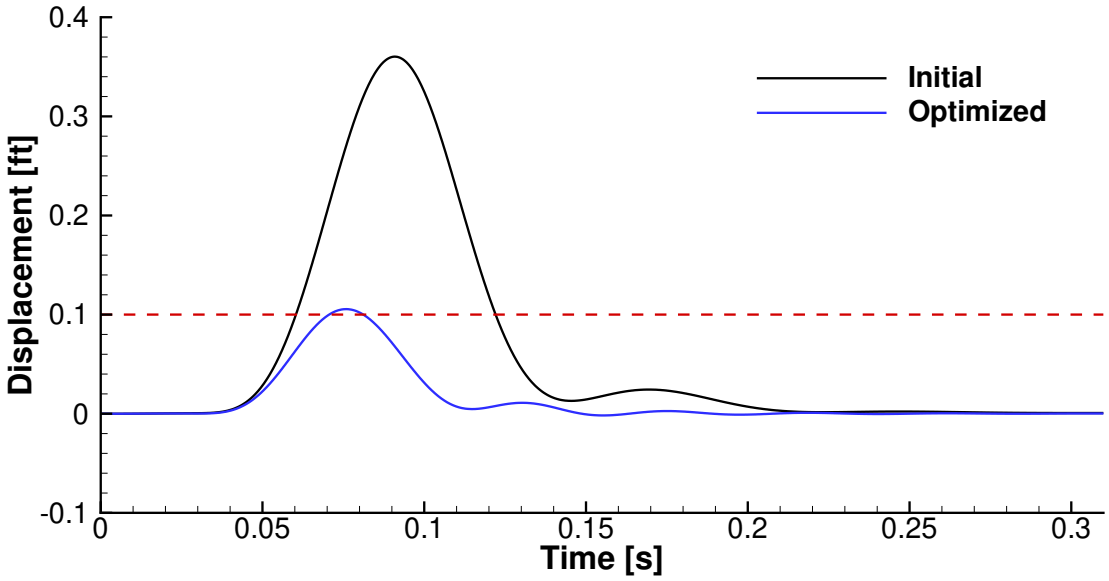


Figure 4.17: Displacement history the NACA 64A010 gust optimization.

CHAPTER 5
AEROELASTIC OPTIMIZATIONS

With the aeroelastic analysis and sensitivity formulation derived and verified, this chapter presents optimizations performed with this capability. The first optimization is an optimization of the vortex-induced vibration of a cylinder. Next a series of optimizations is presented for the undeformed Common Research Model wing which serves as a representative fixed-wing aeroelastic design problem.

5.1 Vortex-induced Vibrations of a Cylinder

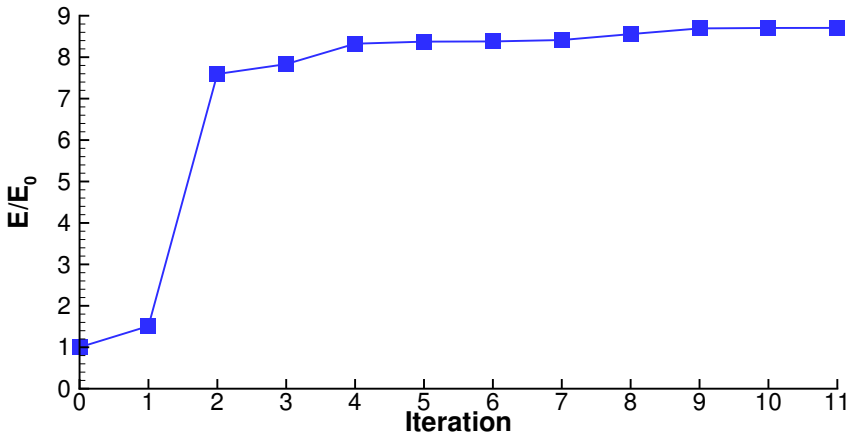
An optimization problem was formulated for the vortex-induced vibrations of a cylinder verification case from Section 3.3.1. Since the vibration of the cylinder is extracting energy from the flow, the problem can be thought of as a simple energy harvesting system where the objective is the maximize the energy extracted by the damper. For this problem, the design variables were chosen to be the spring stiffness and the damping coefficient. This case serves as demonstration that the unsteady aeroelastic adjoint system works on a simple optimization problem (unconstrained optimization with two design variables), but the problem still includes complex fluid-structure interaction physics (interaction of the vortex shedding and the cylinder motion).

Compared to the verification case in Section 3.3.1, several changes were made to the problem setup. Mesh 1 from the mesh refinement study was used and the Reynolds number based on diameter was fixed at 120. The incompressible flow was approximated as a compressible flow at a free stream Mach of 0.1. Only the pressure contributions were included in the force transfer because the required sensitivity terms have only been implemented for the pressure forces in FUN3D. In most aerospace applications, the viscous forces have a negligible effect on the structural deformation. Each simulation in the optimization was

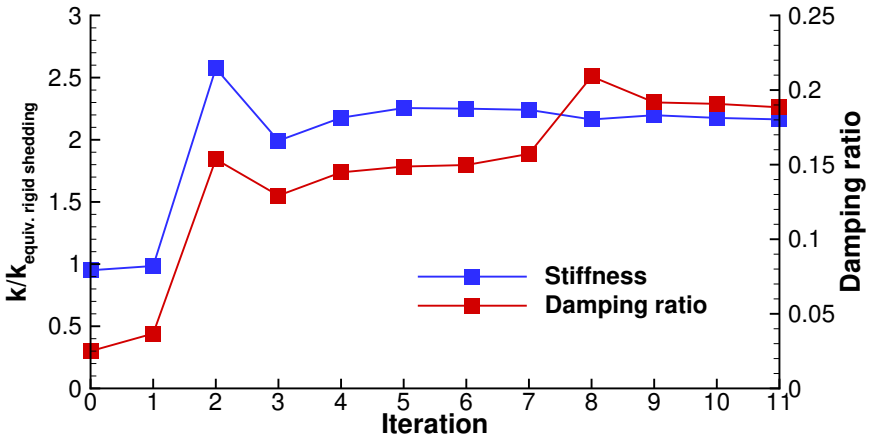
run for 10,000 time steps with approximately 200 time steps per oscillation and 20 subiterations per time step. The objective function only included energy harvested over the final 3,000 steps to avoid the effects of initial transients. The energy harvested was calculated by integration of the power:

$$E = \int_{t_1}^{t_2} c\,\dot{z}^2 dt. \tag{5.1}$$

The initial stiffness was selected such that the rigid shedding frequency and natural frequency of the spring were approximately equal. The initial damping ratio was the experimental damping from Reference [129]. Figure 5.1a shows the history of the objective function during the optimization. Utilizing the SLSQP optimizer, the optimization converged after 11 design cycles (the relative change in the objective is 0.017% for the final iteration), and the final energy harvested was 8.7 times larger than the initial value. Figure 5.1b shows how the design variables varied over the optimization. The damping ratio increased from 0.025 to 0.188, and the stiffness of the spring increased so that the natural frequency of the optimized system is 1.47 ($\sqrt{2.16}$) times higher than the fixed cylinder vortex shedding frequency. Figure 5.2 compares the initial and optimized displacement time-history for the window over which the objective function is defined. In both cases, the cylinder is locked-in since there is only one frequency in the motion. With the optimized design parameters, the higher natural frequency creates faster oscillations. Despite having a smaller amplitude, the optimized design has more oscillations from which to extract energy, and the higher damping ratio means that more energy is extracted per oscillation. The optimizer has taken advantage of the fact that motion of the cylinder affects the vortex shedding, and increasing the motion frequency can also increase the shedding frequency as illustrated by the closer spacing of the vortices in Figure 5.3. The effect of the cylinder's motion on the shedding behavior allows the system to remain locked-in despite the natural frequency being about 1.5 faster than the rigid shedding frequency.



(a) Objective function.



(b) Design variables.

Figure 5.1: Energy harvesting optimization history for the vortex-induced vibration of a cylinder.

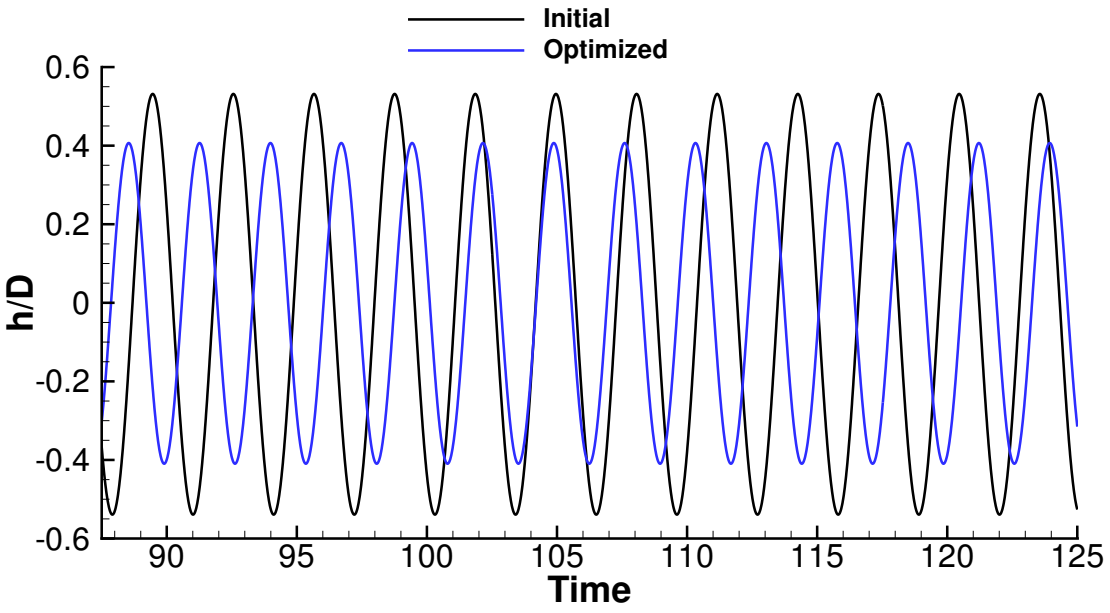


Figure 5.2: Displacement history over the time range that the objective function is defined in the vortex-induced vibration optimization.

5.2 Undeformed Common Research Model

The NASA Common Research Model (CRM) is a conventional transonic aircraft configuration developed for computational verification experiments [156]. Different versions of the CRM are utilized as test cases in the AIAA CFD Drag Prediction Workshop [157] and the AIAA CFD High Lift Prediction Workshop [158]. The original CRM model is assumed to be rigid with predeformed wings that approximate aeroelastic deflections for a 1g cruise condition. The undeformed Common Research Model (uCRM) is an aeroelastic model reverse engineered from the original CRM [66]. The uCRM geometries include a representative wingbox structure and jig shape outer mold line. Planform properties of the two uCRM models are given in Table 5.1. The uCRM outer mold line is shown in Figure 5.4, and an exploded view of the wing box is given in Figure 5.5. The wingboxes includes leading edge and trailing edge spars, ribs, and the upper and lower skins between the spars. The different colors in Figure 5.5 represent the panels which can have different thicknesses in the structural model.

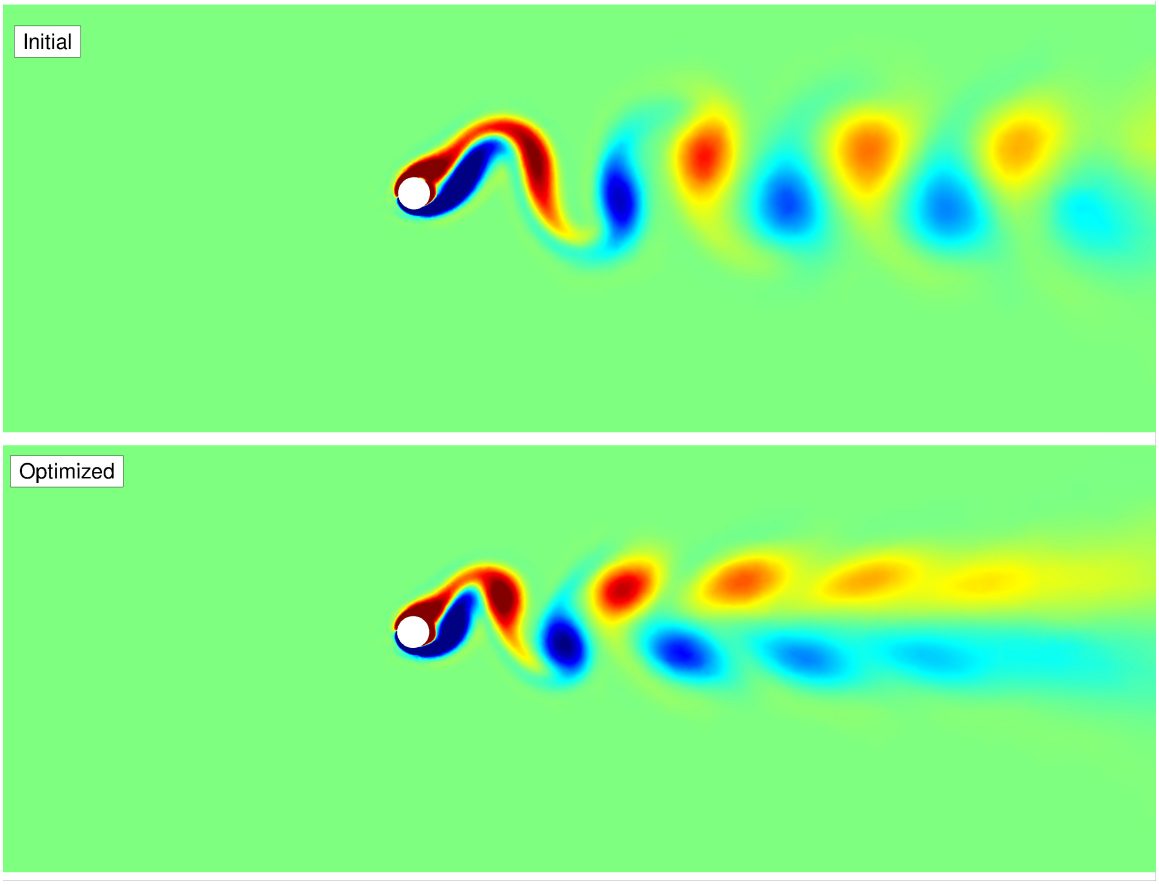


Figure 5.3: Comparison of vortex wake for the initial and optimized vortex-induced vibration case.

Table 5.1: uCRM properties [113].

Property	Value
Aspect ratio	9.0
Span [m]	58.76
Mean aerodynamic chord [m]	7.01
1/4 chord sweep [°]	35.0
Number of Ribs	49

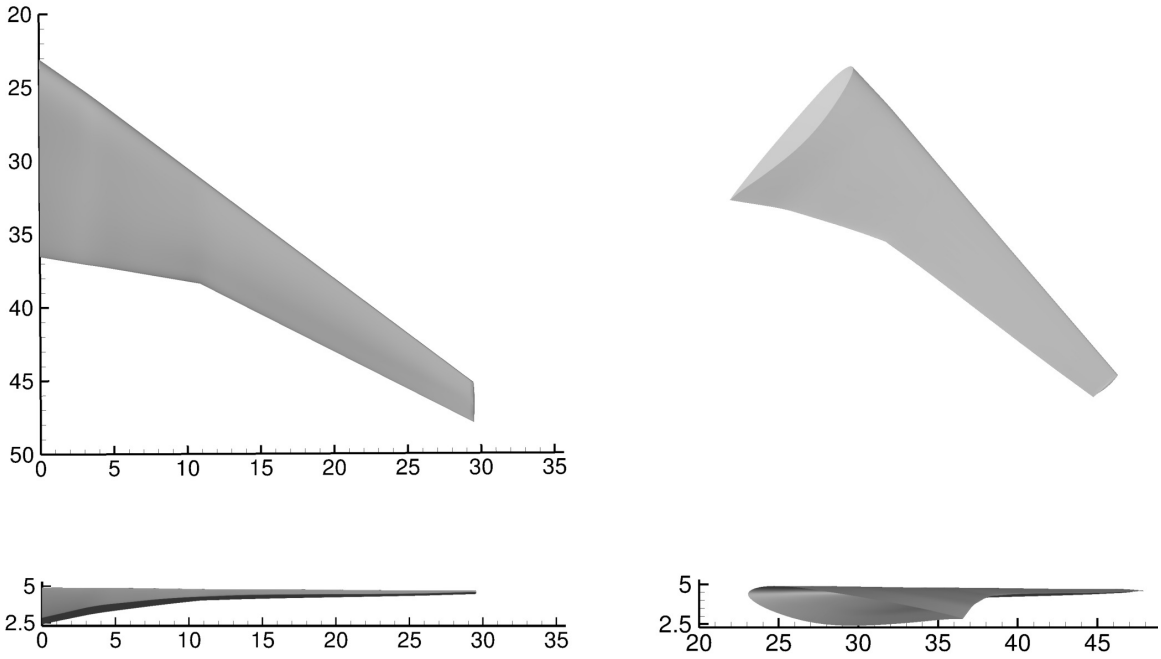


Figure 5.4: uCRM outer mold lines.

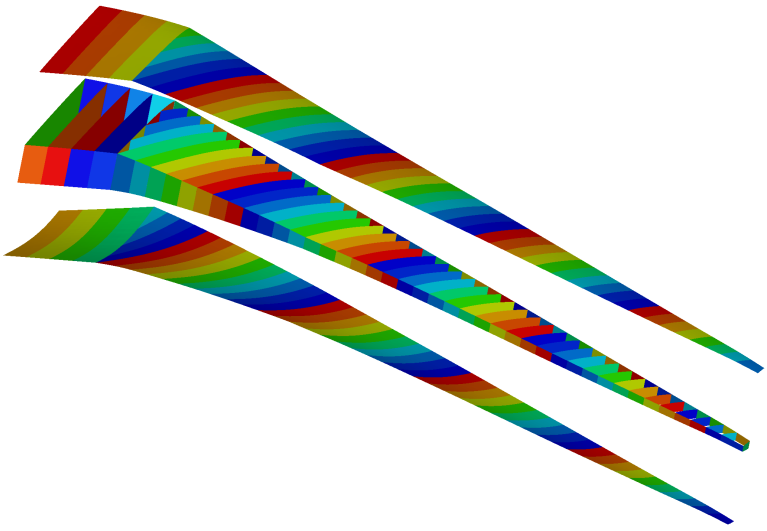


Figure 5.5: Exploded view of uCRM wingbox panels. Each panel indicated by the colors has an associated thickness design variable.

The uCRM wing is a good case to study the effectiveness of the proposed coupling methodology on a more realistic aerospace problem. Three optimizations are presented for the uCRM wing. The first optimization is a steady-state aeroelastic optimization which minimizes the estimated takeoff gross weight of the aircraft. The optimization is subject to stress constraints from a 2.5G pull-up maneuver simulated with steady-state analysis. In the second optimization, the steady problem was modified to have the wing undergo plunging. This was the first optimization performed with the time-accurate coupled adjoint in order to test its capabilities in a design context. Finally, the steady-state takeoff gross weight minimization is presented again but with steady RANS CFD and an additional stress constraint from a 1-cosine gust.

While these optimizations are more aerospace-like design problems applications, they are not complete aircraft optimization problems. In more complete optimizations as envisioned in the CFD 2030 grand challenge problem #3, there would be many more load conditions and cruise performance conditions, more disciplines like controls, and a more detailed representation of the aircraft's structure and inertia. While the physics be interpreted in the following sections as part of evaluating the success of the optimization, the focus of these uCRM optimizations is to demonstrate that the mathematics and methodology can be applied to three-dimensional aerospace configurations. Therefore, the computational models in the following sections are coarser than typically utilized to accurately analyze the aeroelastic system to reduce the required computational resources.

5.2.1 Steady Takeoff Gross Weight Optimization

The first optimization is a takeoff gross weight (TOGW) minimization of the uCRM wing. For the steady optimization, performance data in Table 5.2 is estimated from the Boeing 777-200ER similar to Reference [67] since it is roughly the same size as the uCRM. Each design evaluation in the optimization is comprised of two steady simulations: a cruise and maneuver condition. The cruise Mach number is 0.85 at an altitude of 35,000 ft. Load lim-

Table 5.2: Assumed CRM conditions [66]

Parameter	Value
Cruise Mach number	0.85
Cruise altitude	35,000 ft
Maneuver Mach number	0.86
Maneuver altitude	20,000 ft
Maneuver load factor	2.5
Design range	7,725 nm
Thrust-specific fuel consumption	0.53 lb/(lbf h)
Maximum takeoff weight	298,000 kg
Reserve fuel weight	15,000 kg
Fixed weight	108,900 kg

its on the wing are governed by the maneuver condition which is a 2.5G symmetric pull-up modeled at a Mach number of 0.86 at 20,000ft. Four optimizations were performed using Euler and RANS aerodynamics with and without shape design variables. The optimal design from the Euler analysis without shape variables was utilized as the initial configuration for the Euler optimization with shape variables as well as for the RANS design without shape variables. The optimal design from the Euler optimization with shape variables was then the initial condition for the corresponding RANS problem.

Computational Model

The structural mesh consists of 10,584 mixed interpolation of tensorial components (MITC) shell elements [159] which model the ribs, spars, and upper and lower skins. The linear MITC shell elements in TACS are given material properties of a 7000 series aluminum alloy. The inertia of the engine and the fuel in the wing was ignored in the problem. Both Euler-based and RANS-based optimizations were performed. The aerodynamic grid for the Euler-based gust simulation was a tetrahedral mesh consisting of 60,742 nodes. The RANS mesh was 412,910 nodes with an initial normal spacing such that $y^+ \approx 4$, and the SA turbulence model was utilized. The MELD transfer scheme transferred the coupling data with 200 structural nodes in the subset of nodes attached to each aerodynamic surface

Table 5.3: uCRM steady design variables.

Variable Type	Quantity
Angle of Attack	2
Structural thickness	240
Twist	9
Camber	70
Total	321

node, and the exponential decay parameter β set to 0.5.

Design Variables

The design variables in the optimization problem include structural variables, aerodynamic variables, and shape variables which affect both the structural and aerodynamic models. The structural design variables are the thicknesses of components of the wing's ribs, upper and lower skin, and the two spars. There are 240 total thickness variables as illustrated in Figure 5.5a, in which each panel is governed by a separate structural thickness variable. The aerodynamic variables are the angles of attack at the cruise condition and at the maneuver condition. The shape variables are chosen from a parameterization of the geometry using MASSOUD [121] in Figure 5.6. Nine twist variables (the root twist was fixed to avoid redundancy with the angle of attack variable) and 70 camber control points allowed the optimizer to change the shape of the wing while maintaining the same planform area and span.

The optimizations were attempted with more shape parameterization freedom in MASSOUD including planform and thickness design variables; however, more sophisticated geometry parameterization and/or mesh generation is necessary. With the aerodynamic outer model line shape and the wingbox tied to the same geometry parameterization, the optimizer tried to reduce the size of the wingbox near the tip which created a poor quality CFD mesh. Figure 5.7 illustrated how the optimizer's preference to reduce the wingbox size and weight adversely impacts the CFD mesh. Additionally, the chordwise MASSOUD con-

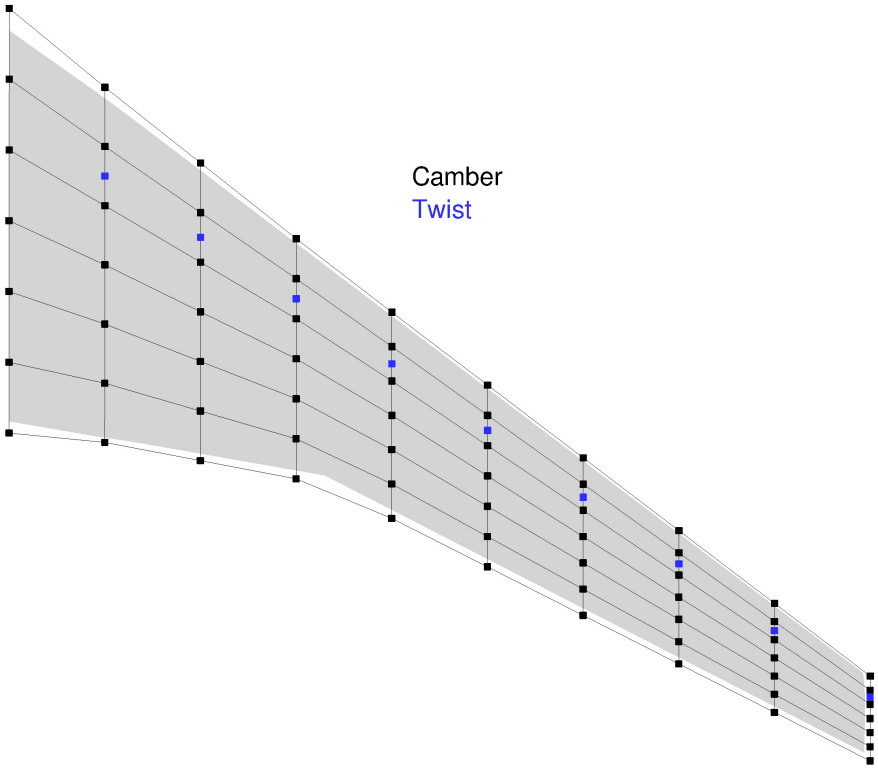
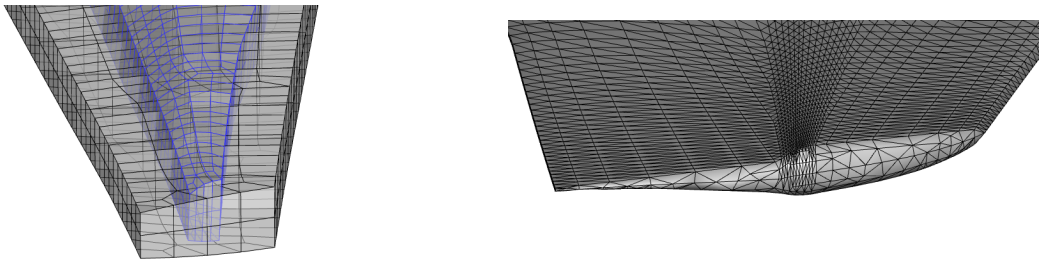


Figure 5.6: Shape parameterization control points for the uCRM wing.



(a) Reduction of the wingbox.

(b) Effect of the wingbox reduction on the CFD surface mesh.

Figure 5.7: Effect of single parameterization on the uCRM wing geometry during TOGW minimization.

trol points have been pushed towards the midchord in order to reduce the wingbox's size, so there are insufficient control points to adjust the shape of the downstream half of the wingtip cross section. This problem can be alleviated by individual shape parameterization of the aerodynamic surface and the internal structure. Separate parameterizations would permit changes of the wingbox inside the outer skin without influencing the CFD mesh quality, but it is not a trivial task to implement as there must be additional constraints such as a requirement that the wingbox does not protrude from the outer skin.

For the Euler-based optimization without shape variables, the initial values of the angle of attack for the cruise and maneuver conditions were zero degrees. The structural components were given an uniform thickness of 20 mm, and the initial values of the shape design variables are zero, i.e., the given shape of the undeformed CRM wing. The optimized Euler-based design without shape variables was the initial design for the RANS optimization without shape variables and the Euler optimization with shape variables. The Euler-based design with shape variables was the initial condition for the corresponding RANS optimization.

Objectives and Constraints

The objective of the optimization is to minimize the takeoff gross weight (TOGW) of the CRM aircraft. For this problem, the fuel burned during takeoff, climb, and landing are

neglected. Therefore, the weight of the aircraft at the beginning of the cruise segment is taken as the maximum takeoff gross weight. The maximum takeoff gross weight is found by rearranging the Breguet range equation

$$TOGW = W_2 e^{(Rc_T/V_\infty)(D/L)}, \tag{5.2}$$

where R is the range, c_T is the thrust specific fuel consumption of the engines, V_∞ is the speed of the aircraft, D is the drag, and L is the lift. W_2 is the weight of the aircraft at the end of the cruise segment. It is assumed to take the following form:

$$W_2 = 2W_{wing} + W_{fixed} + W_{reserve\ fuel}, \tag{5.3}$$

where the weight of the wing is a function of the design variables.

A summary of the constraints in the optimization problem are given in Table 5.4. Due to manufacturing considerations, smoothness constraints are placed on the thickness variables so that adjacent panels on the spars, upper skin and lower skin vary by less than 1 mm. There are 368 total smoothness constraints (two linear constraints per pair of adjacent sections). To prevent failure of the wing at the 2.5G maneuver, a KS failure constraint which is the KS aggregation of the von Mises stress to yield stress ratio is placed on the structure with a material safety factor of 1.5. There are two trim constraints that ensure that the wings produce enough lift to equal the weight of the aircraft for the cruise condition and 2.5 times the weight for the maneuver condition. The weight in the trim calculations is the average of the TOGW and W_2.

The objective and constraints in this design problem are simplified compared to what would be necessary for a complete vehicle. For example, the constraints do not consider the internal volume of the wing available for fuel, the trim constraints do not include trim of the pitching moment or account for the forces and moments of the fuselage and stabilizers, and the drag in the TOGW calculation does not include components apart from the wings.

Table 5.4: Steady uCRM optimization constraints.

Constraint Type	Equation	Quantity
Smoothness of structural thickness	$\pm(x_i - x_j) \leq \delta$	368
Structural failure for maneuver	$1.5 \times KS \leq 1$	1
Trim at cruise	$W - L = 0$	1
Trim at maneuver	$2.5 \times W - L = 0$	1
Total		371

Table 5.5: Individual simulation times in the maneuver and gust uCRM optimizations.

Scenario	Aero. model	CPU Cores	Forward [min]	Adjoint [min]
Cruise	Euler	80	0.75	4.3
	RANS	240	9	32
Maneuver	Euler	80	0.75	4.3
	RANS	240	9	32

To compute the gradients, four adjoint solutions are required: lift and drag for the cruise condition; and lift and KS failure for the maneuver condition. The smoothness and planform constraints are purely functions of design variables, and their gradients are evaluated analytically. The computational cost for each of the analyses required to compute the objectives, constraints, and their gradients is summarized in Table 5.5. Even with four times as many CPUs, total wall time required for the maneuver design cycle (all the forward and adjoint solutions) increases from 10.1 minutes for the Euler version to 82 minutes for the RANS version. Although the scenarios were simulated sequentially in this work, both scenarios could simulated simultaneously to reduce the wall time per design cycle.

Optimization Results

The optimizations were performed with the SLSQP optimizer in pyOpt. Figure 5.8 shows the history of the optimizations. The infeasibility is a measure of the constraint violation and is calculated as the L-2 norm of the active constraints. The optimizations were considered converged with the TOGW changing less than 0.001% between design cycles and the

infeasibility was less than 0.01. With this level of infeasibility, the KS aggregated stress ratio was within 0.0005 of 2/3 (the maximum allowable value with the safety factor), and the lift constraints were within 200 N (approximately 0.015% of the TOGW) of being satisfied.

For the Euler-based optimization without shape variables, there was a large drop in TOGW over the 15 design cycles as the optimizer thinned out the structure. However, the infeasibility increased over this period as the stress in some panels exceeded the allowable stress ratio. Therefore the TOGW increased over the next seven design cycles to reduce the stress ratio, and the infeasibility dropped by more than an order of magnitude. Over the remaining 65 design cycles, the optimizer refines the mass distribution to further reduce the TOGW and constraint violation. For the initial RANS design without shape variables, there was a discontinuity of 0.7% in the TOGW because RANS model predicted more drag than the Euler model for the same configuration. Because of the change in predicted load distribution between the Euler and RANS models, the initial RANS design did not violate the stress constraint, but the lift lift constraints were initially violated. For the first RANS design cycle, the optimizer adjusted the angles of attack to satisfy the lift constraints as indicated by the large drop in infeasibility. It then decreased the panel thicknesses until the stress reached the maximum allowable value again. The RANS optimization without shape variables converged in only 18 design cycles because the optimal Euler-based configuration was near the optimal RANS configuration in the design space. Despite running 18 design cycles compared to 87 Euler design cycles, the RANS optimization's computational cost was approximately five times larger. The final RANS design's TOGW was 0.3% higher than the Euler design without shape variables. Allowing the optimizer to modify the camber and twist distribution of the wing reduced the TOGW by 2.3% and 2.0% for the Euler and RANS designs respectively. As with versions without shape variables, initializing the RANS optimization from the final Euler design meant that the RANS optimization converged quickly. It is expected that include more freedom to control the shape would

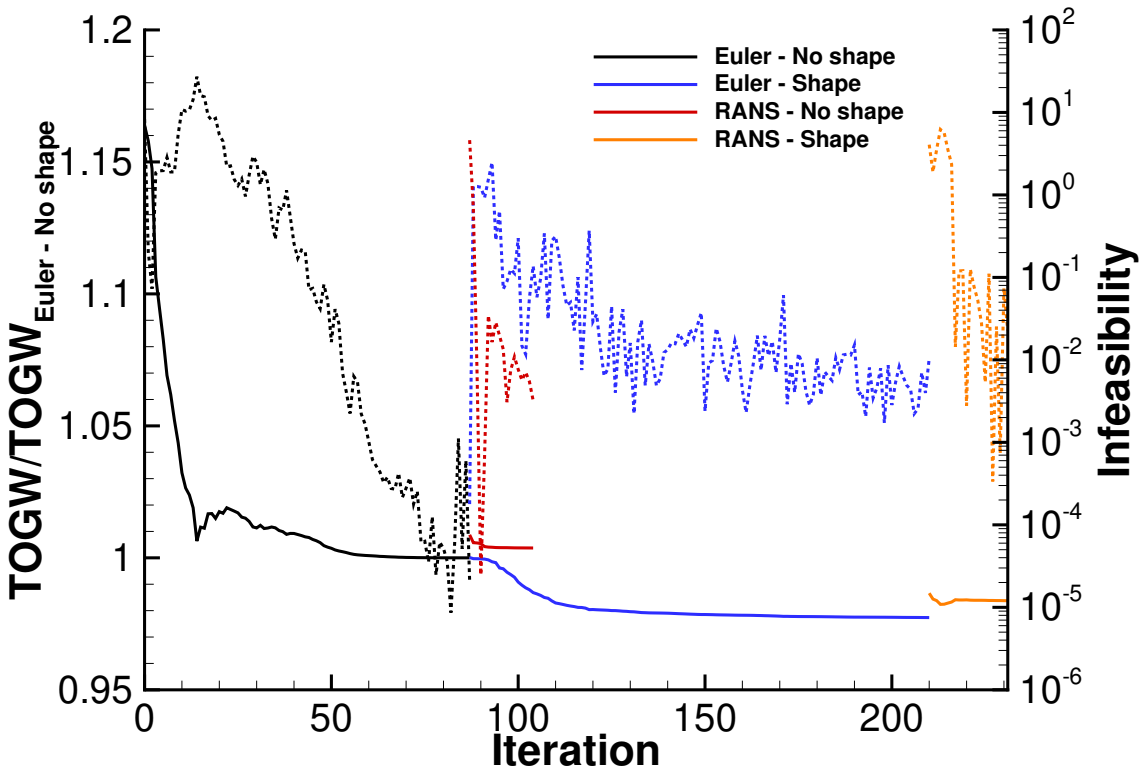


Figure 5.8: Steady uCRM optimization history.

allow the optimizer to further reduce the TOGW.

There are two ways in which the TOGW in Equation 5.2 can be reduced by the optimizer. It can decrease the weight W_2, which is a function of the wing weight, or it can increase the lift-to-drag ratio. The histories of these values through the optimization are illustrated in Figure 5.9. The optimizer prioritized weight reduction as a means of reducing the TOGW as indicated by the decrease in both wing weight and the lift-to-drag ratio. When the shape variables were introduced to the Euler design, there was an initial increase in lift-to-drag ratio, but it then dropped in order to further decrease the mass of the wing. When switching from the Euler to RANS aerodynamic models, there is a discontinuity in the lift-to-drag ratio which increased the TOGW as reflected in Figure 5.8.

Figure 5.10 depicts the thickness of the structure for the four optimal designs, and Figure 5.11 gives the ratio of von Mises stress to maximum allowable stress for the structure. The maximum stress ratio occurs at the junction where the trailing edge spar changes

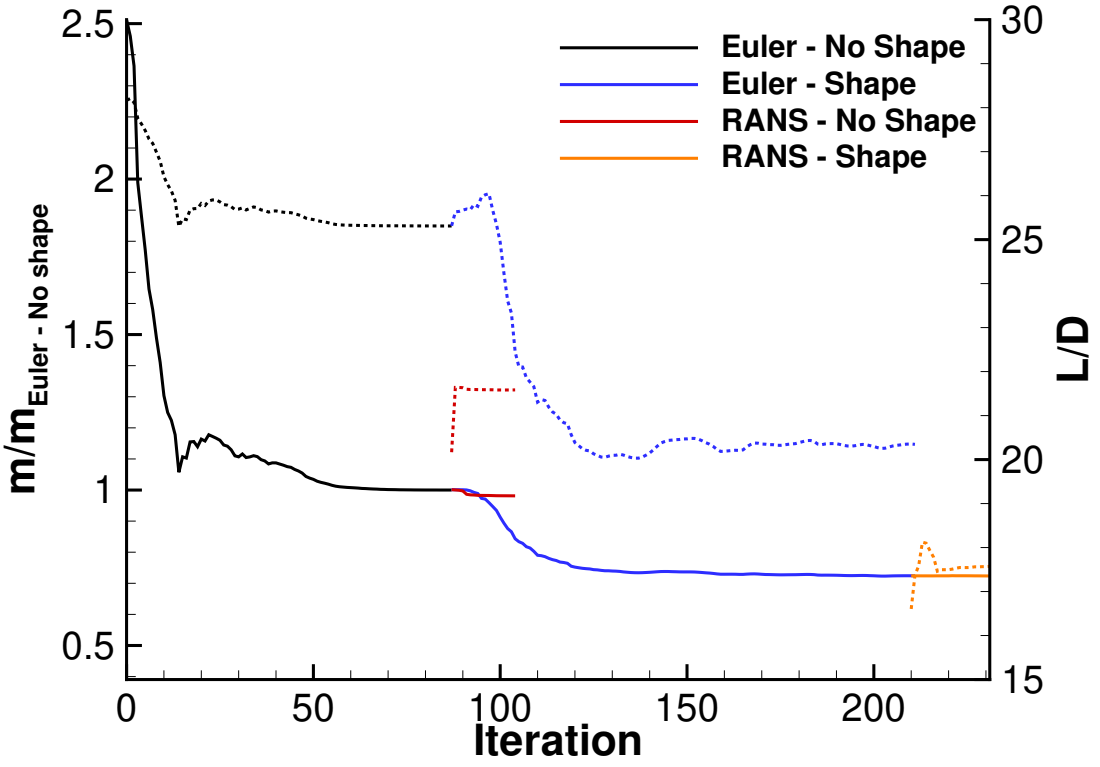
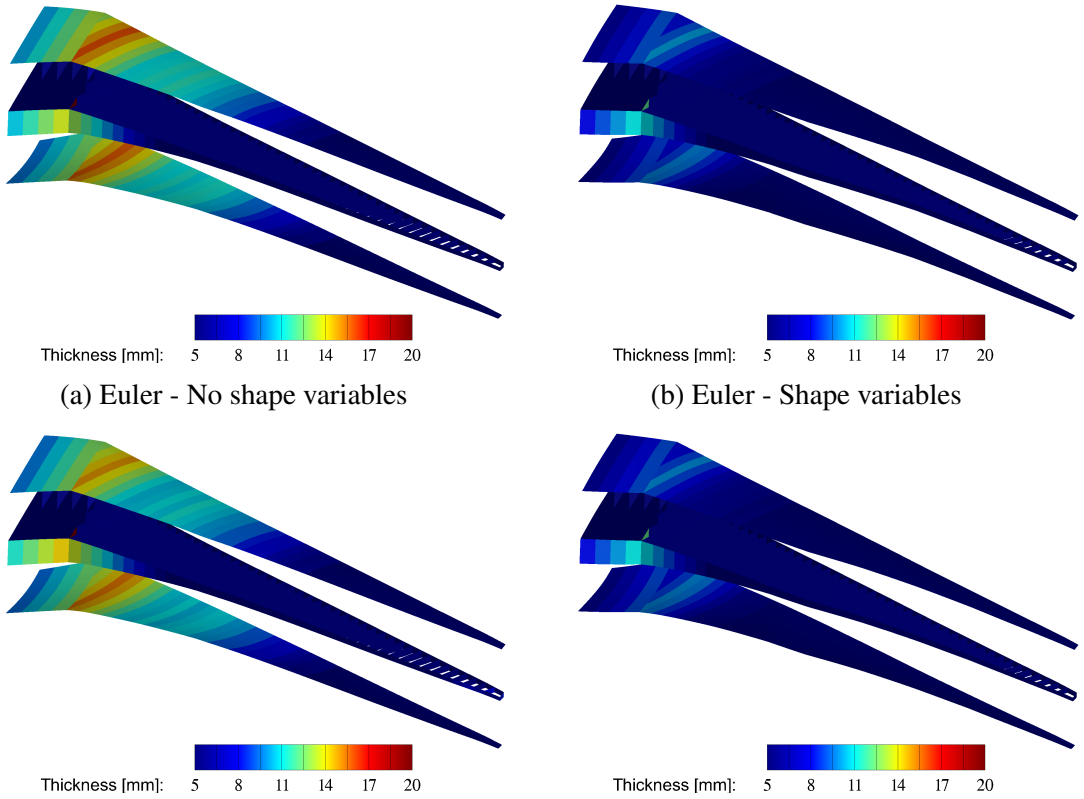


Figure 5.9: Steady uCRM history of normalized wing weight (solid lines) and lift-to-drag ratio (dotted lines) during the maneuver-constrained optimizations.



(a) Euler - No shape variables

(b) Euler - Shape variables

(c) RANS - No shape variables

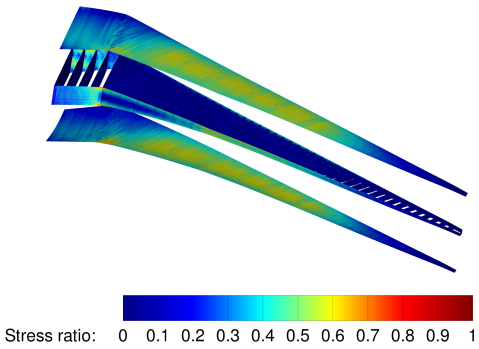
(d) RANS - Shape variables

Figure 5.10: Thickness of the structural panels for various optimized designs

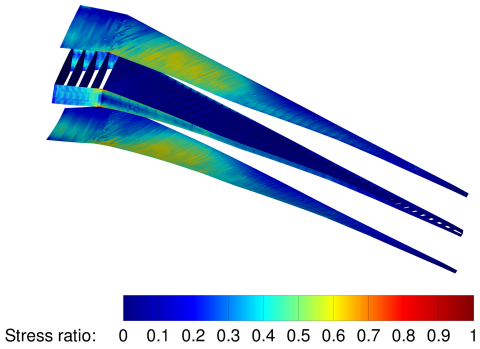
.

direction and the fifth rib is attached. The thickest panels of the wingbox are therefore concentrated around that region. These panels are about 4 mm thinner in the designs with shape variables. Apart from the panels at the design variable lower bound of 5 mm, the RANS-based wingbox without shape variables have skin panels that are on average 1 mm thinner, but spars that are 0.8 mm thicker compared to the Euler-based design. For the designs with shape variables, the differences in panel thickness are less than a 0.3 mm. This leads to approximately the same wing weight.

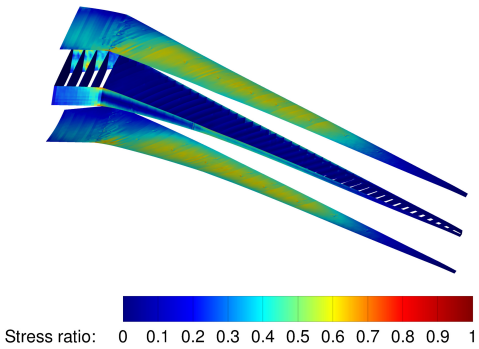
The regions of higher stress ratios in the outer half of the wings without shape variables is eliminated in the designs with shape control. This is caused by the lift distribution shifting inboard as illustrated in Figure 5.12. In this plot, the bell-shaped lift distribution represents Prandtl's optimal lift distribution for a given vehicle weight [160], and the ellip-



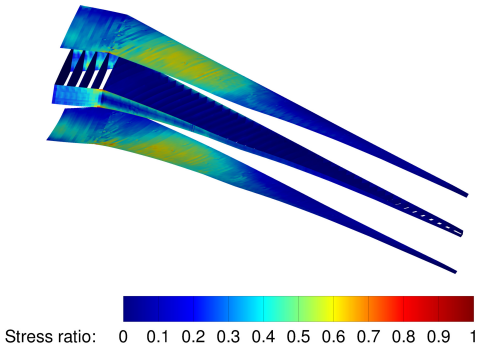
(a) Euler - No shape variables



(b) Euler - Shape variables



(c) RANS - No shape variables



(d) RANS - Shape variables

Figure 5.11: Stress ratio (von Mises stress/maximum allowable stress) for the initial and optimized uCRM wing at the 2.5G maneuver condition.

tical distribution is the most efficient lift distribution for a given wing span. The designs without shape variables produce lift distributions near the optimal lift distributions, but the addition of shape freedom shifted the inboard to reduce the bending moment at wing root which in turn reduces stress at the trailing edge spar junction. In fact, the optimized designs with shape variables produce negative lift at the wing tip to further reduce the bending moment. The focus to reduce the bending moment rather than creating a more efficient lift distribution is consistent with Figure 5.9 where the reduction of weight was observed as the preferred method of TOGW minimization for the optimizer. One method that the optimizer utilized to achieve this inboard shift in lift distribution by increasing the jig shape washout as illustrated in Figure 5.13. As depicted in Figure 5.14, the optimizer has increased the camber of the inboard sections to increase the inboard lift and gradually adds negative camber as the span station increases to reduce the outboard lift. As with the thickness distributions, the Euler and RANS jig shapes were very similar. The most significant difference is the camber near the trailing edge in the outer spans of wing where the RANS-based design has decreased camber at the trail edge because the RANS-based optimizer must consider that large camber will lead to separated flow and increased drag.

The takeoff gross weight optimizations produced unconventional airfoil designs and lift distributions. This illustrates why a complete problem definition is important is automated design optimization. Otherwise, the algorithm will exploit any gaps in the problem definition and violate unstated constraints. In this case, one such unstated constraint could be that the designer does not want loss of aerodynamic performance during the TOGW minimization.

5.2.2 Plunging Wing Optimization

The next optimization was the first case attempted with the implemented time-accurate methodology; therefore, Laminar flow was assumed to reduce the computational cost compared to a RANS simulation and mesh. The Reynolds number per meter was set to 7,000

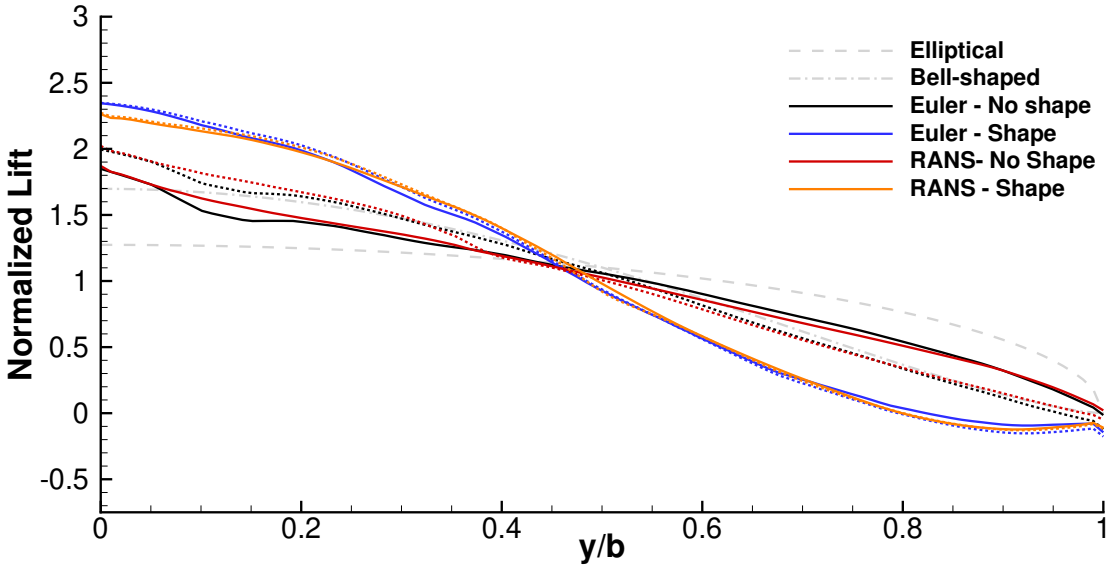


Figure 5.12: Lift distributions for the steady uCRM optimized designs. The solid lines are the cruise lift distributions and the dashed lines are the maneuver lift distributions.

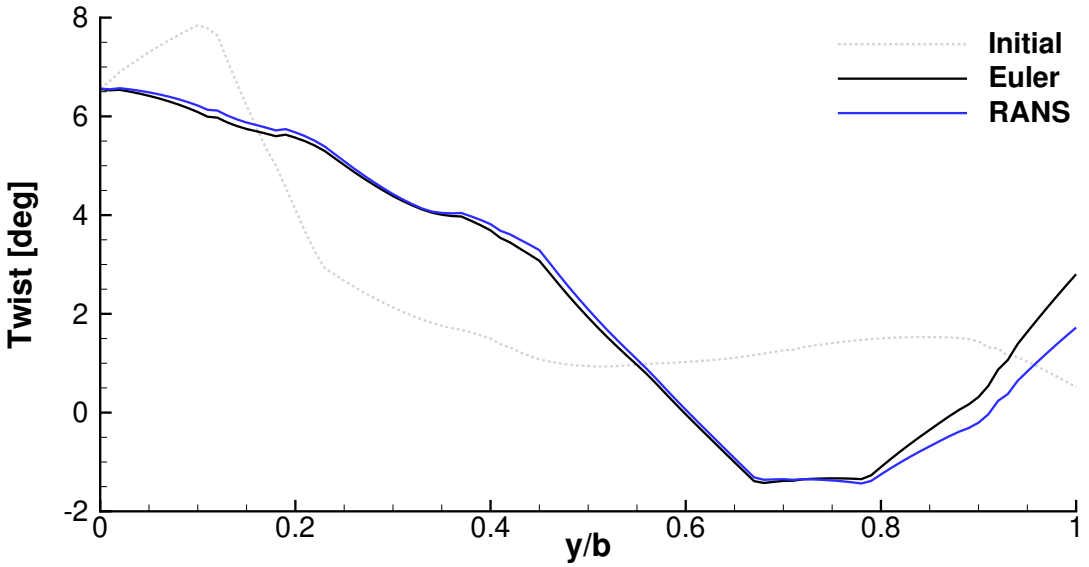
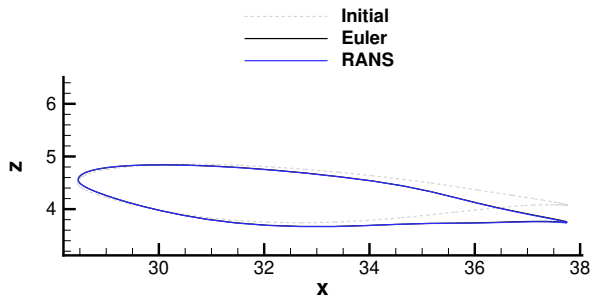
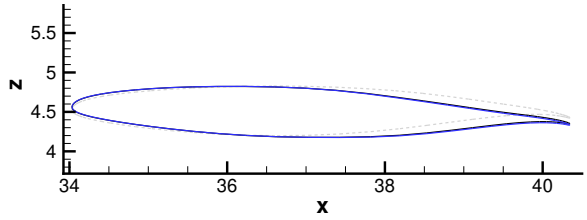


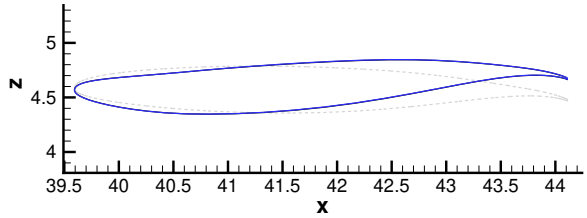
Figure 5.13: Twist distribution in the jig shape for the optimized uCRM designs.



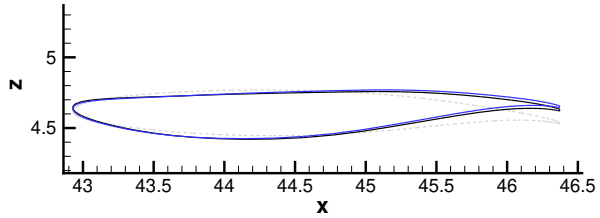
(a) 25%.



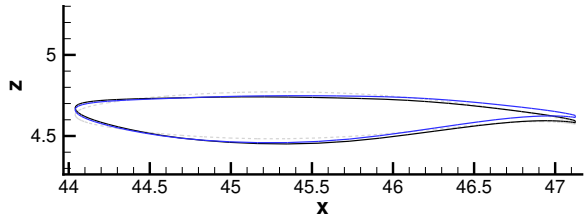
(b) 50%.



(c) 75%.



(d) 90%.



(e) 95%.

Figure 5.14: Wing cross-sections for the different steady optimizations of the uCRM.

or 998.6 per mean aerodynamic chord. The assumed conditions for the uCRM wing are the same as the steady optimization in Table 5.2 except only the cruise condition is analyzed. Rather than a steady cruise condition, the wing root undergoes a sinusoidal plunge motion with an amplitude of 2 cm at frequency of 29.66 Hz which is near the 31.42 Hz natural frequency of the structure. Each simulation was computed for three oscillation cycles, with the objective and constraints defined on the final cycle.

The aerodynamic grid is a mixed-element mesh consisting of approximately 1.3 million nodes with 13,787 surface nodes and approximately 60 nodes in the normal boundary layer. The structural model is identical to the one applied in the steady optimization. The time integration schemes BDF2OPT and BDF2 are used for the aerodynamic and structural models, respectively, with 50 time steps per cycle of driven motion.

Design Variables

The set of design variables is very similar to the steady optimization. The same 240 structural variables controlled the thickness of the wingbox panels. The initial structure has a uniform thickness of 15 mm. There is one aerodynamic variable, the cruise angle of attack which was initially set to zero degrees. There were 48 shape design variables in the MASSOUD parameterization to control the camber and thickness defined at the 24 control points in Figure 5.15. The initial shape is the original jig shape. Very small bounds on the design variables were maintained to ensure that combinations of design variables did not create meshes with negative cell areas during this initial test of the time-accurate coupled adjoint.

Objective and Constraints

As in the steady uCRM optimization, the objective here is to minimize the takeoff gross weight given by Equation 5.2. Because it is an unsteady simulation, the lift and drag in the objective are time-averaged over the final oscillation of the forced motion.

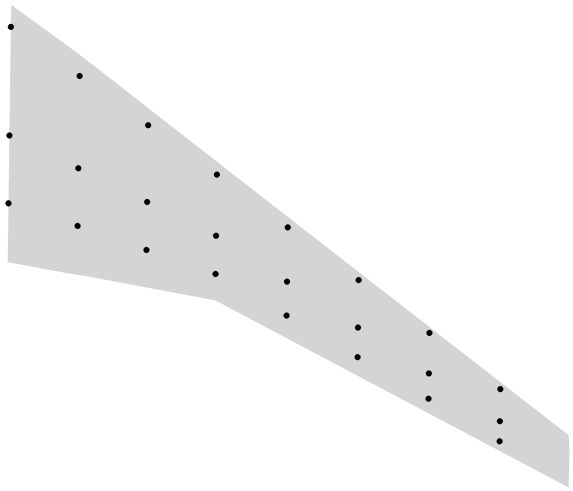


Figure 5.15: Shape parameterization locations for the plunging wing uCRM optimization.

The constraints are summarized in Table 5.6. All but two of the constraints are related to the smoothness of the thickness distribution of the wingbox. These constraints are identical to those in the steady optimization. The next constraint is to ensure that the time-averaged lift over the final cycle is equal to the aircraft weight. The final constraint is the structural stress constraint. The stress constraint was based on the KS aggregation of stress ratio (von Mises stress / yield stress) over the wingbox and the time domain to approximate the maximum stress in the entire simulation:

$$KS(\rho,\sigma) = m + \frac{1}{\rho}\ln\left[\int_{t_1}^{t_2}\int_{\Omega} e^{\rho(\sigma-m)}d\Omega\,dt\right], \tag{5.4}$$

where the value of ρ has been set to 200. This aggregation allows a single differentiable function (and single adjoint) to monitor the maximum stress over the entire spatial and time domain of the problem. The author could not find another example of aggregating stress constraints over space and time like this in the literature. The constraint included a safety factor of 1.5:

$$c(x) = 1.5 \times KS(\rho,\sigma) - 1 \leq 0. \tag{5.5}$$

Table 5.6: Unsteady uCRM optimization constraints.

Constraint Type	Equation	Quantity
Smoothness of structural thickness	$\pm(x_i - x_j) \leq \delta$	368
Lift	$W - L = 0$	1
Structural failure	$1.5 \times KS \leq 1$	1
Total		370

Optimization Results

The optimization was performed for 50 iterations, and the convergence of the optimization is provided in Figure 5.16. The takeoff gross weight was reduced by 10.85%; however, Figure 5.16 indicates that the takeoff gross weight has not converged and could potentially be reduced further. During the first 2 iterations, the optimizer adjusts the angle of attack to trim aircraft, as indicated by the sharp drop in the blue line representing constraint violation. After 50 iterations, the lift constraint has been reduced to approximately 200 Newtons which corresponds to a lift-to-weight balance violation of 0.015% of the TOGW, and the KS failure value is 0.66682 (the constraint requires a value less than or equal 2/3). The violation of the constraints was sufficiently small to consider to the final design feasible (although not optimal since it is not converged).

Figure 5.17a compares the initial and final cross sections at 80% span. The changes are subtle, but the figure indicates that the optimizer has thinned the airfoil near its leading edge to reduce the drag and increased the camber near the trailing edge to increase the lift.

Figure 5.17b shows final distribution of thickness in the structural panels. Like the steady uCRM optimized structure, the optimizer creates thicker panels around the yehudi break to counteract the high stress concentration at the sharp corner; however, the optimized design for the plunging problem also includes thicker panels near the tip. In the initial design in Figure 5.18a, the stress exceeds the allowable amount near the tip because the natural frequency of the mode near the forced motion frequency is excited. This can be observed by comparing the pattern of stress in Figure 5.18a to the mode shape of the closest

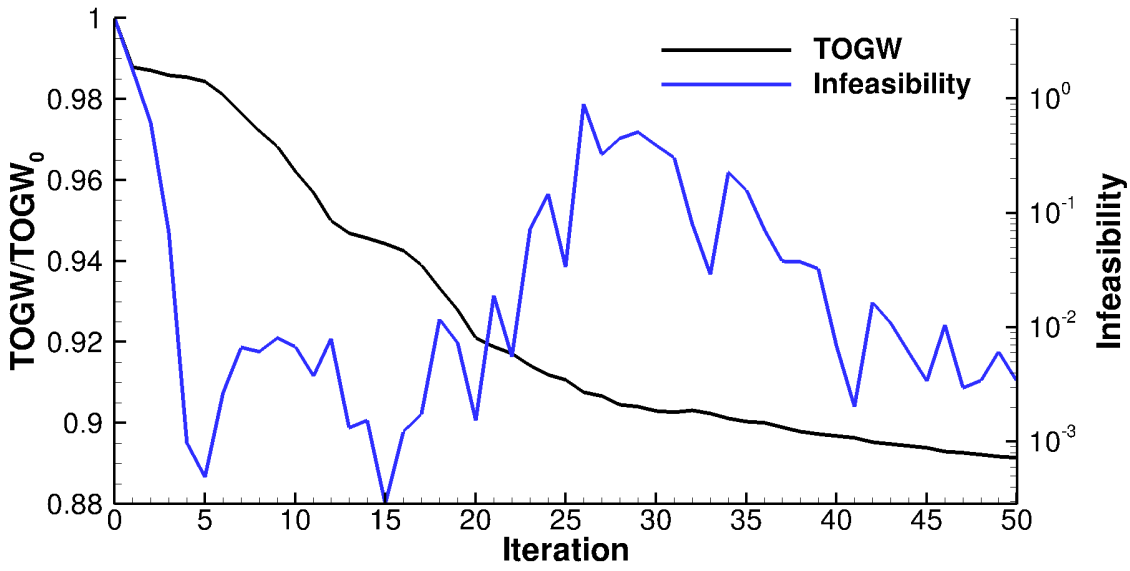
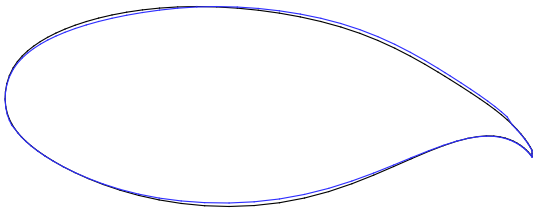


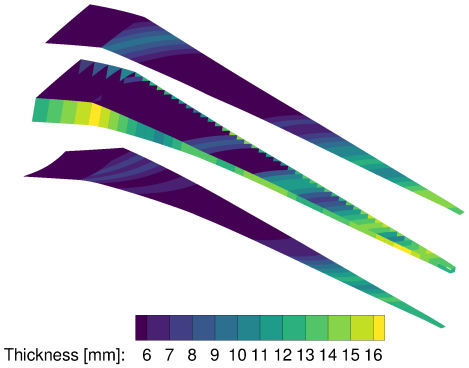
Figure 5.16: Unsteady uCRM optimization history.

natural mode in Figure 5.19a. The optimizer added mass near the anti-nodes of the initial mode shape, so that at the final design, the mode shape nearest to the driven frequency in Figure 5.19b is a chordwise bending mode which is nearly orthogonal to the vertical excitation at the root. This leads to lower stress ratios despite the fact that the panels are thinner near the tip in final design in Figure 5.18b.

Like the steady problem, the optimizer can decrease TOGW by decreasing the empty weight of the aircraft or maximizing the lift-to-drag ratio. The comparison of the lift-to-drag ratio in Figure 5.20 identifies that the final ratio is an average of 4.50% higher than its initial value over the design interval. This increase in the lift-to-drag ratio accounts for 0.74% of the reduction in the takeoff gross weight. The optimizer also significantly reduced the weight of the wing. The final empty weight is 10.11% of its initial value with the wing weight reduction of 53.07%. Nearly all of the reduction in TOGW has been achieved by reducing the empty weight of the aircraft.



(a) Jig cross section comparison at 80% span. Vertical axis exaggerated to show detail. The black curve represents the initial design, and the blue curve represents the final design.



(b) Final panel thicknesses. The initial thickness was a uniform 15 mm.

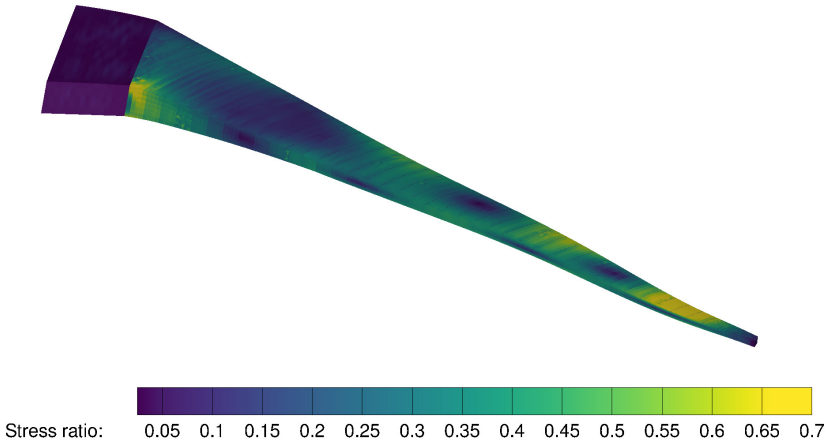
Figure 5.17: Changes in the design variables as a result of the unsteady optimization.

While a sinusoidal plunging wing during cruise is not representative of a real aircraft, it does demonstrate that the unsteady aeroelastic adjoint-based sensitivities can successfully optimize a built-up lifting surface with stress constraints, transonic aerodynamics, and hundreds of design variables.

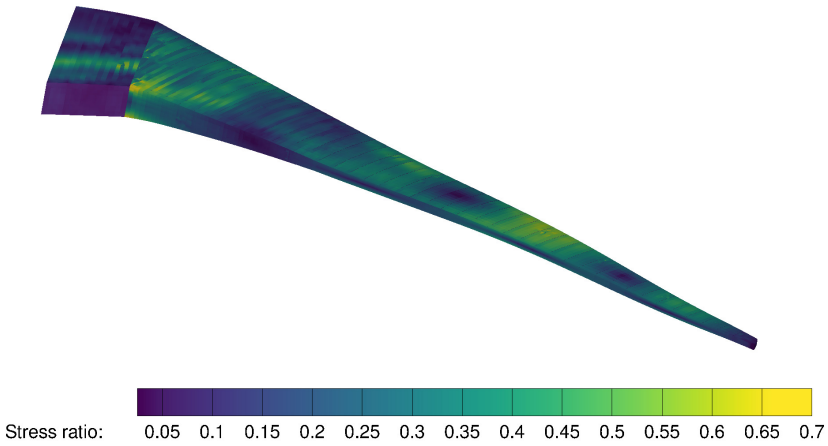
5.2.3 Gust-constrained Mass Minimization

Design Problem

The final pair of uCRM optimizations compares designs based on time-accurate Euler and RANS CFD. The objective of the optimizations was to minimize the mass of the uCRM wing subject to a stress constraint from a gust simulation. The stress constraint was based on the KS aggregation of the stress ratio in Equation 5.4 with a safety factor of 1.5, and the gust characteristics are described below. The design variables in the optimization were the thicknesses of the 240 panels in Figure 5.5. The initial thicknesses were set to a uniform value of 5 mm for the Euler model and 10 mm for the RANS model. The higher thickness in the initial RANS design was to ensure that the very flexible wing did not produce negative volumes under deflection since the RANS mesh has smaller elements that are more likely to invert. In addition to the KS-based stress constraint, the 368 smoothness constraints for the panel thicknesses from Section 5.2.1 were applied. Since these constraints are

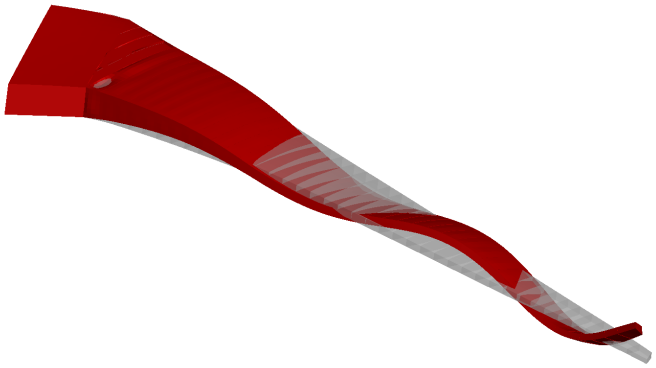


(a) Initial design.

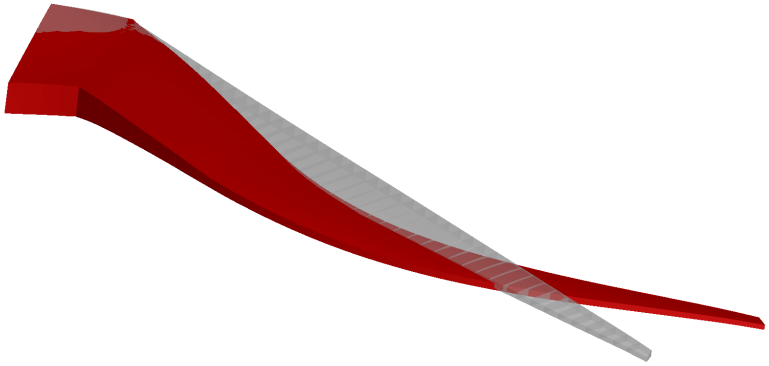


(b) Final design.

Figure 5.18: Comparison of the stress ratio (von Mises to yield stress) in the structure at $t = 2.42T$.



(a) Initial design, $f = 31.42$ Hz.



(b) Final design, $f = 30.05$ Hz.

Figure 5.19: Comparison of the mode shapes nearest to the driven motion frequency, $f = 29.66$ Hz.

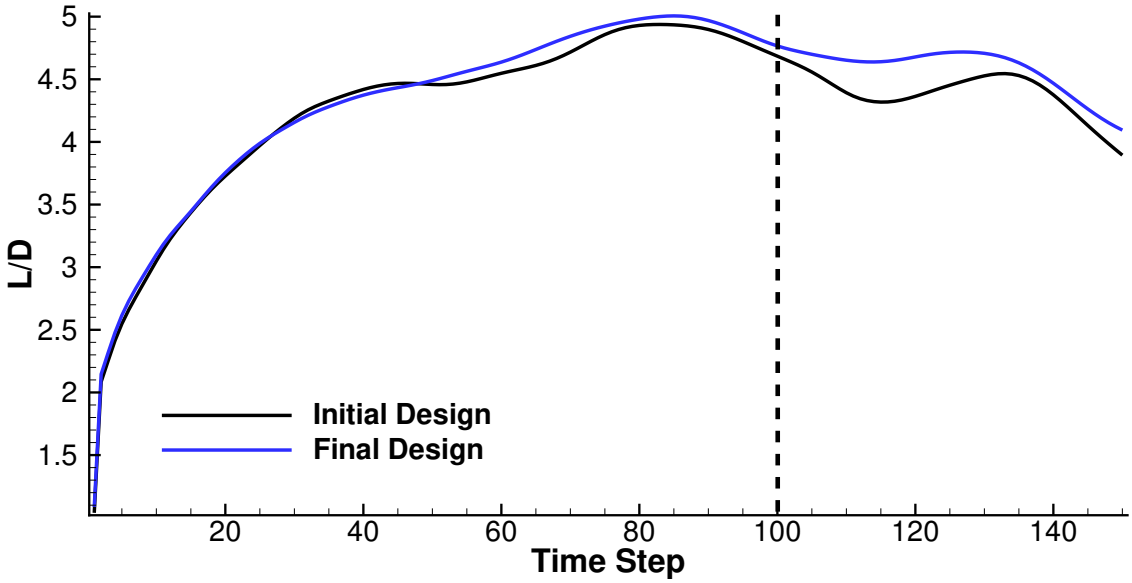


Figure 5.20: Comparison of the L/D ratio. The objective functions and constraints are defined from time steps 100 through 150.

direct functions of the design variables and not the states, they do not require an adjoint evaluation. The mass is also directly a function of the design variables, so this design optimization required one adjoint solution per design point for the KS aggregate stress ratio.

Computational Model

Each gust simulation consisted of 500 time steps with approximately 42 steps per gust period. The angle of attack was fixed at 5 degrees. The computational meshes were those described in Section 5.2.1. In FUN3D, 30 subiterations per time step were applied to converge the pseudo-time residuals at least five orders of magnitude at each time step. The 1-cosine gust simulation conditions in Table 5.7. The altitude and Mach number are those of the cruise condition in Section 5.2.1. The flight profile alleviation factor was calculated based on the properties of a Boeing 777 and the equations described in Section 4.2. An example lift history is depicted in Figure 5.21. The simulations began from rest at free-stream conditions. Over the first 250 time steps the dynamic pressure in the load transfer is linearly varied from zero to the full dynamic pressure. This eliminated large dynamic stresses due to the sudden application of lift and allowed the wing to reach approximately static equilibrium before encountering the gust. As the leading edge of the wing root encountered the gust at around 1.25 seconds, a sudden variation in the lift was experienced by the wing. For the Euler model, the wing experienced an increase in lift as the gust created a larger instantaneous angle of attack. However in the RANS model, the gust caused flow separation on the upper surface of the wing in Figure 5.22a, and the lift dropped. The reverse flow region quickly shed in Figure 5.22b as the gust passed, and lift was recovered. The rapid variation in lift caused the wing to oscillate in both cases. Enough time was simulated to ensure that the maximum stress was captured as the oscillations damped out. For the Euler analysis, each forward and adjoint evaluation required 12 and 20 minutes, respectively, on 80 cores. For the RANS analysis, the forward and adjoint evaluations took

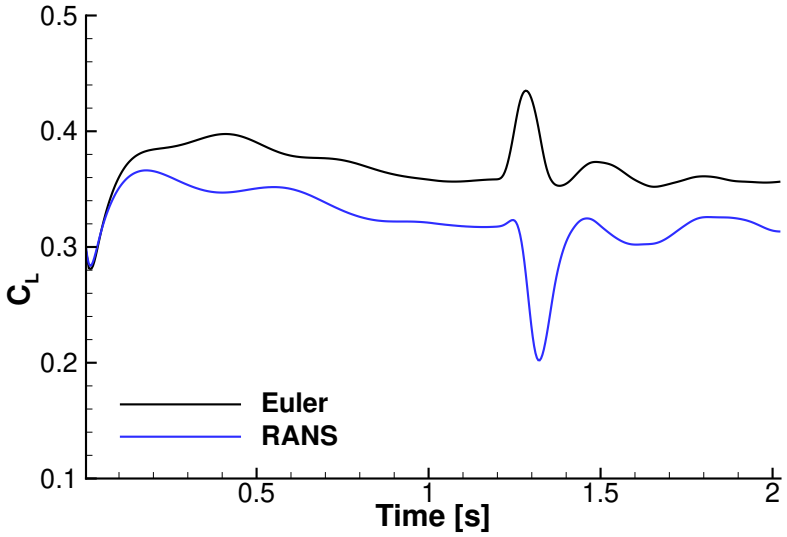


Figure 5.21: Lift coefficient during the uCRM gust simulation.

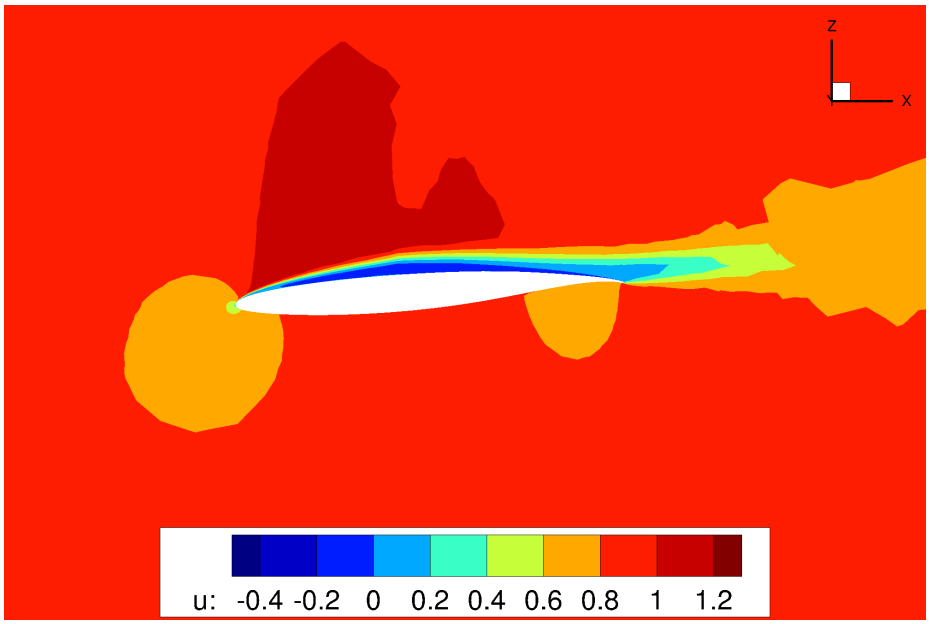
Table 5.7: Assumed conditions for the uCRM gust constraint.

Property	Value
Altitude [ft]	35,000
Free-stream Mach number	0.85
Flight profile alleviation factor, F_g	0.812
Gust gradient distance, H [ft]	30.0

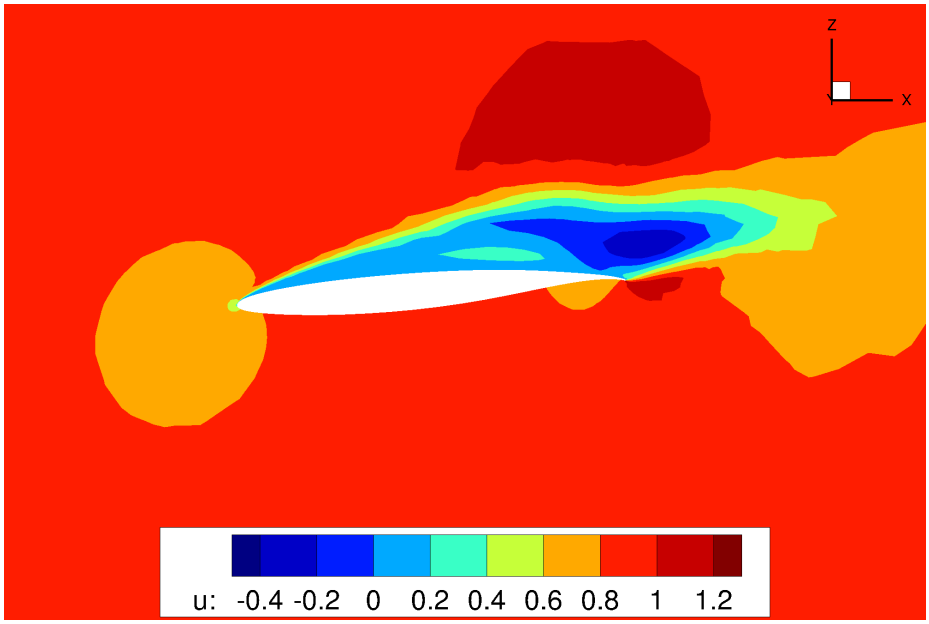
40 and 130 minutes respectively on 240 cores.

Optimization Results

The optimizations were performed with the SLSQP optimizer. The Euler and RANS optimizations converged in 30 and 23 iterations, respectively, as illustrated in Figure 5.23. At the end of both optimizations, the mass changed by less than 0.01% per design iteration, and the KS aggregated stress ratio is within 1.0×10^{-5} of the constraint limit of 2/3 indicating that the optimizations have converged. The gusts caused the initial wings to exceed the safe stress level at the sharp corner of the trailing edge spar and both skins in Figure 5.24; the failing region of the Euler simulation is significantly larger primarily do to the thinner initial wing panels. The optimizer increased the thickness of panels in the failing regions resulting in heavier wings as the optimization converged as illustrated in Figure 5.23. The



(a) $t = 1.29$ s.



(b) $t = 1.33$ s.

Figure 5.22: Contours of u velocity at $y/b = 0.8$ for the RANS-based gust analysis. The dark blue region indicates reverse flow created by the gust.

mass of the optimized Euler wing is 1.84 times heavier than the optimized RANS wing. Much of the extra mass in the Euler wing is in the upper skin and ribs in the region of the sharp corner in the trailing edge spar to accommodate the stress concentration there. In the Euler analysis, the steady-state lift in Figure 5.21 was higher, and the gust increased lift. This combination led to a larger root bending moment and trailing edge stress concentration compared to the RANS analysis. In the RANS analysis, the highest stress occurs as the wing bounces back after the sudden loss of lift as indicated in Figure 5.26c where the highest stresses occur around $t = 1.618$ s when the discrete gust had already convected downstream of the wing. In the Euler analysis, the highest stress occurs around $t = 1.416$ s near the peak lift created by the gust.

These optimizations demonstrated that the proposed methodology is capable of aeroelastic design optimization with time-accurate RANS analysis. While Euler and RANS models produced very similar designs in the steady optimization in Section 5.2.1, the unsteady analyses produced very different physics when the discrete gust was encountered. In this problem, the Euler design is conservative, but in general that may not be the case. For example, at the Mach numbers above 1.0, the Euler analysis of the AGARD 445.6 wing in Figure 4.11 is not conservative compared to the RANS analyses. Therefore, it is important to include viscous effects in the analyses of these complex, unsteady physics problems.

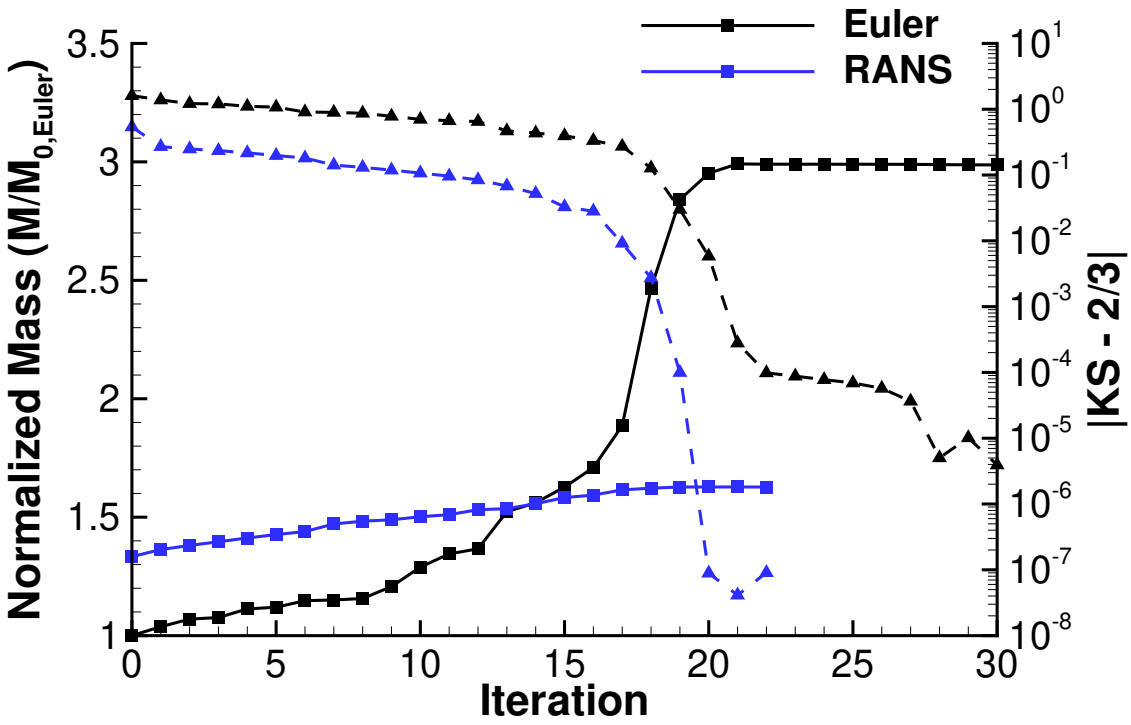
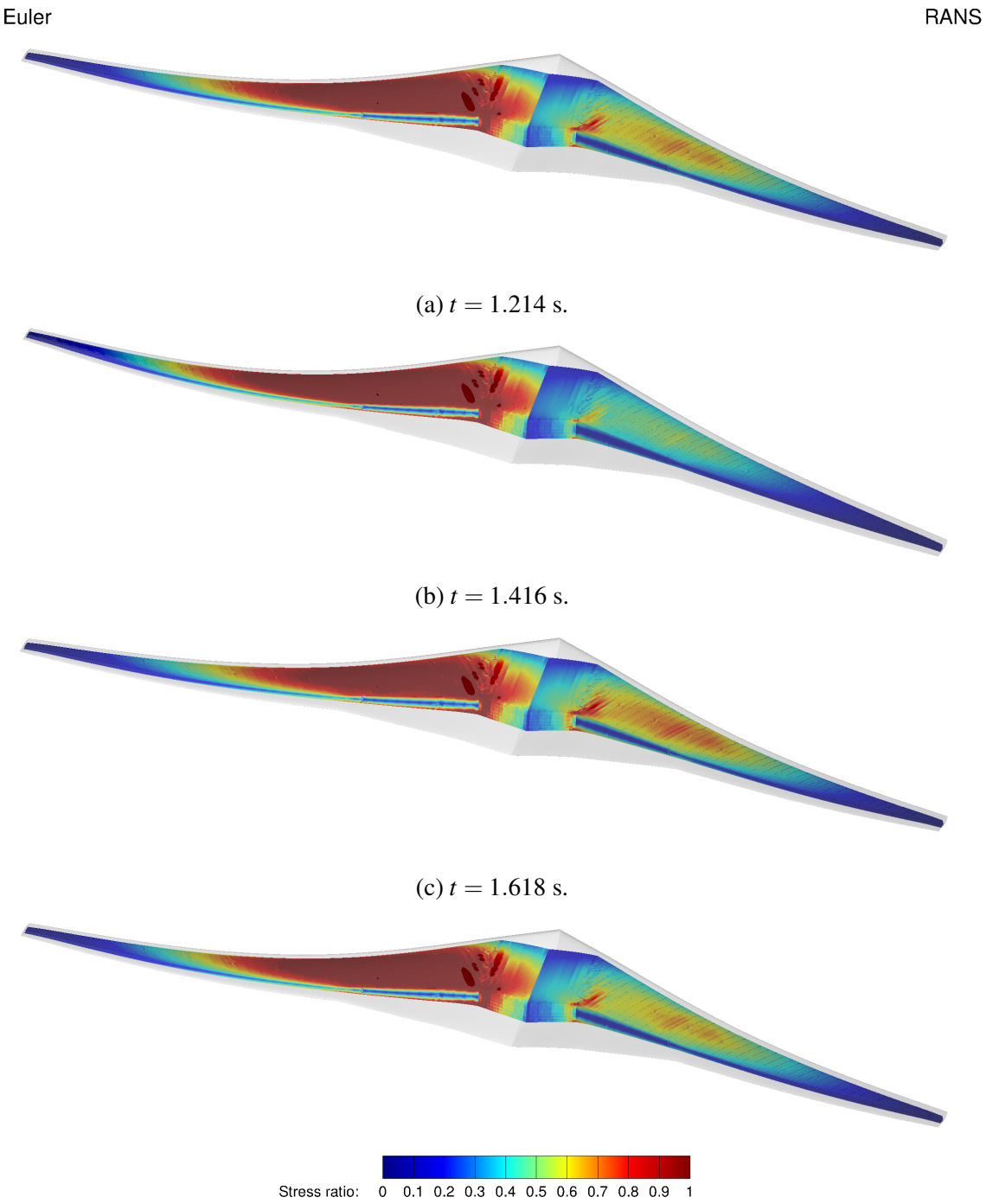


Figure 5.23: Optimization history of the uCRM mass minimization. The solid lines with squares are the normalized wing mass, and the dashed lines with triangles are the absolute value of the difference between the aggregated stress ratio and the constraint limit.



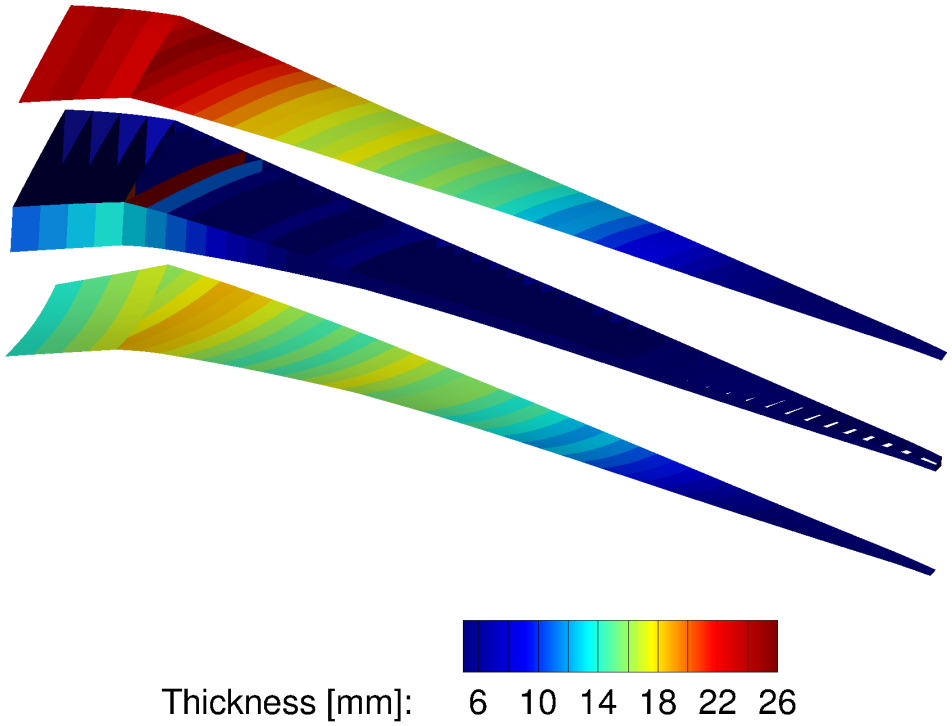
(a) $t = 1.214$ s.

(b) $t = 1.416$ s.

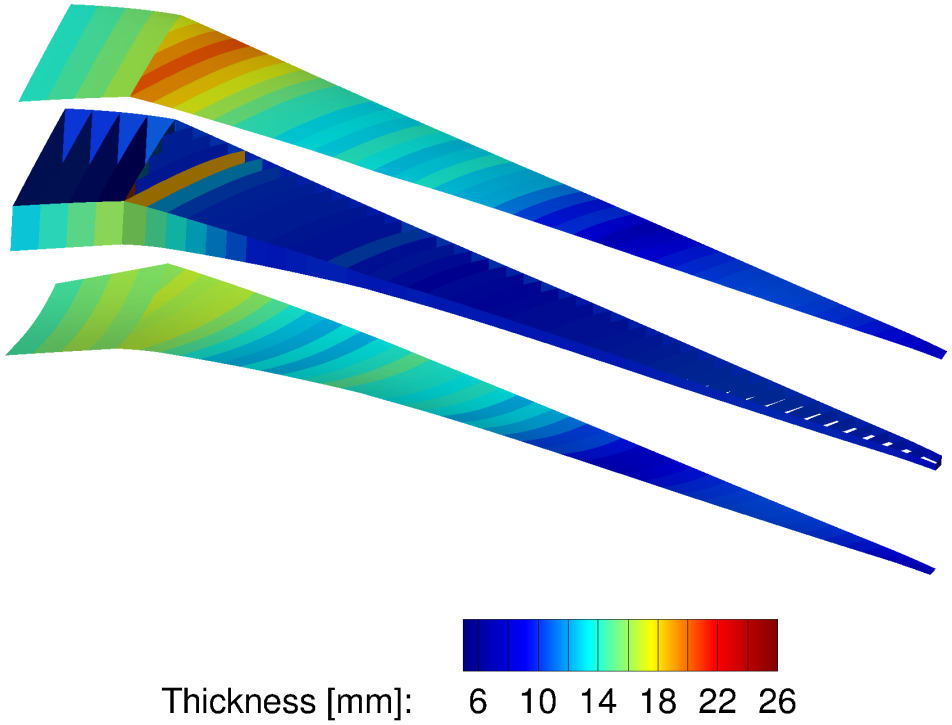
(c) $t = 1.618$ s.

(d) $t = 1.820$ s.

Figure 5.24: Stress ratio (von Mises stress to yield stress) for the initial uCRM wings with Euler and RANS aerodynamics.

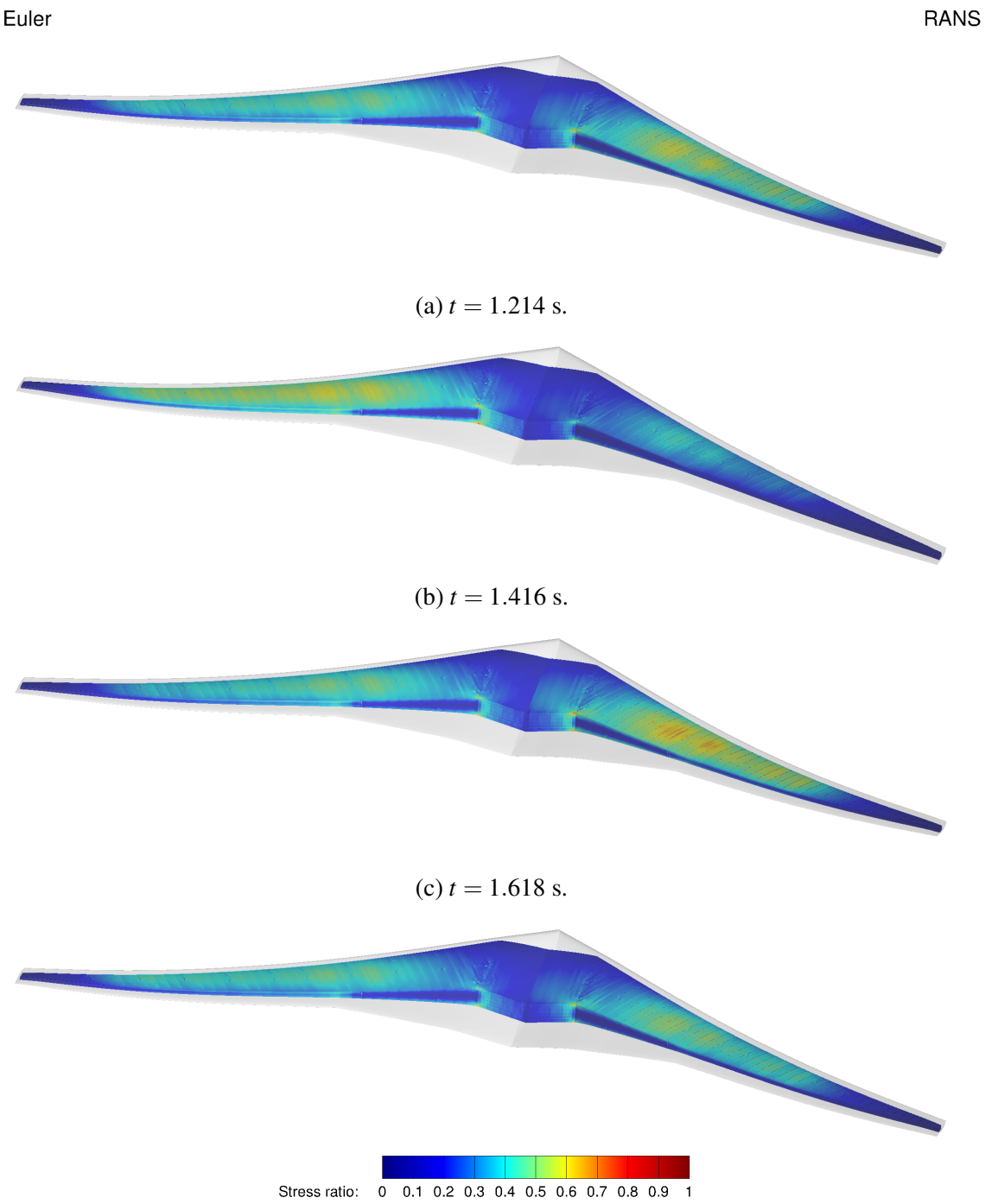


(a) Euler.



(b) RANS.

Figure 5.25: Optimized thickness distribution for the uCRM mass minimization.



(a) $t = 1.214$ s.

(b) $t = 1.416$ s.

(c) $t = 1.618$ s.

(d) $t = 1.820$ s.

Figure 5.26: Stress ratio (von Mises stress to yield stress) for the optimized uCRM wings with Euler and RANS aerodynamics.

CHAPTER 6
CONCLUSIONS

The present work advances the state of the art for multidisciplinary analysis and optimization in the following ways:

- **A more general aeroelastic coupling strategy for adjoint-based design**

  The proposed framework includes steady and time-accurate analysis as well as a discrete adjoint implementation for coupled CFD and structural FEM. The high-fidelity analysis captures nonlinear behavior in the analysis. It allows structural, aerodynamic, and aeroelastic constraints in the optimization problem formulation. Large motion and deflections are captured with a combination of rigid and deforming motion in the displacement transfer scheme and the CFD analysis. The structural model can vary from simple spring-mass-damper systems to built-up finite element models. Many of these aspects have been demonstrated individually in the literature; however, this research presents a more unified framework that allows more complex aeroelastic optimizations to be performed.

- **Time-accurate flutter and gust constraints for CFD-based analysis**

  Utilizing the coupling framework developed in this research, time-domain constraints for flutter and gust response have been developed and demonstrated. For flutter, the matrix pencil method has been confirmed as the most robust system identification method for determination of the damping of a response in the context of time-accurate CFD-based analysis. Constraints based on the matrix pencil method were applied to identification of flutter conditions for a NACA 64A010 airfoil with pitch and plunge degrees of freedom and the AGARD 445.6 wing. Gust response constraints were illustrated with discrete gusts modeled with the field velocity method.

This method was applied to a mass minimization of the uCRM wing subject to a gust-based stress constraint.

The aeroelastic coupling framework was utilized in an optimization for the vortex-induced vibration of a cylinder. This energy harvesting problem illustrated that the time-accurate adjoint-based optimization is capable of optimization in complex fluid-structure interaction environments such as the lock-in of the cylinder motion and vortex shedding.

Optimizations of the uCRM wing served as application of the methodology to a more realistic aerospace configuration. The optimizations included design variables for aerodynamics, structures, and shape; and objectives and constraints from steady and time-accurate aeroelastic analysis.

6.1 Recommendations for Future Work

- **Improved Geometry Parameterization**

  As discussed in Section 5.2.1, driving both the internal structural components and outer mold line shape from the same geometry parameterization can lead to poor quality meshes. It can also lead to impractical designs. For example, the designer may want to keep the internal wingbox structure cross section rectangular for manufacturing considerations but add more camber to the outer mold line. With all of the parts of the wing tied to the same free-form shape parameterization, this is not possible. Integration of the developed aeroelastic capability with advanced geometry and meshing tools such as EGADS [161] may allow for more control over the shape parameterization and more freedom to robustly adjust the shape and planform.

- **Computational Efficiency and Multifidelity Optimization**

  While this work has demonstrated that unsteady adjoint-based aeroelastic optimization is viable, the computational cost and memory requirements remain an impediment to its widespread use. While the code was written to operate in a parallel

environment, no effort has been dedicated specifically to reducing the computational cost. In addition to improvement of the current algorithm, there are opportunities to reduce the computational costs and make the implementation more efficient with alternative coupling algorithms and reduced memory requirements through checkpointing. Even with these improvements, the cost will be driven up as more design points/constraints are considered in order to analyze the full envelope of the design. One approach for future use of the time-accurate coupled adjoint is to consider it as a component within the full range of tools discussed in the literature review rather than a replacement for those tools. With this mindset, the time-accurate analysis and sensitivities would be applied in situations where the other tools are insufficient while utilizing the less expensive tools whenever possible. For example within the same design problem, an optimization could have steady CFD analysis for cruise conditions, frequency domain analysis for flutter stability over the flight envelope, a panel method for continuous gust response at subsonic conditions, and a time-accurate CFD-based analysis for transonic gust response. The same idea could be applied to the structural side, e.g., a structure could be represented with equivalent beam models or modal analysis when the deflections are important but not the stress (potentially in cruise conditions) and with more detailed FEM in the scenarios where higher fidelity is necessary, such as analyzing the stress due to the gust response or supersonic panel flutter. A sequencing of the physics and model fidelity could also be done to reduce the overall cost, e.g., utilizing a optimized gust-constrained design based on a panel method as the initial design in a time-accurate CFD-based optimization. This would be similar to the sequencing of the Euler and RANS CFD in Section 5.2.1 where the majority of the optimization was performed with the less expensive Euler model, and smaller adjustments were made with the more expensive RANS computation.

- **Adaptation and Error Estimation**

  Although coarse meshes can be useful for the initial stages of an optimization, it is

important to ensure that the answer is accurate at the final design cycle. Rather than manually generating a sequence of meshes, mesh adaptation could automate the mesh sequencing process and more efficiently refine the meshes. Since the adjoint solution is already calculated for design sensitivities, it would be available for adjoint-based adaptation and error estimation [29, 108, 162]. Mesh adaptation could occur either every design cycle to more rapidly get an accurate simulation or less frequently to allow more design changes with coarser meshes.

- **Additional Disciplines**

  While very important, aerodynamics and structures are not the only disciplines that need to be considered. The adjoint formulation presented in Chapter 3 was developed with the intention of extending it to include more disciplines. Potential disciplines to consider next could be thermal effects which are critical for electric aircraft and hypersonic vehicle design, acoustics which are important for all aircraft, particularly new urban air mobility concepts, or controls which could permit design of active flutter suppression systems or inclusion of stability and control surface effectiveness constraints.

REFERENCES

[1] E. Dowell, J. Edwards, and T. Strganac, "Nonlinear Aeroelasticity," *Journal of Aircraft*, vol. 40, no. 5, pp. 857–874, 2003.

[2] E. Livne, "Future of Airplane Aeroelasticity," *Journal of Aircraft*, vol. 40, no. 6, pp. 1066–1092, 2003.

[3] J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. Mavriplis, "CFD Vision 2030 Study: a Path to Revolutionary Computational Aerosciences," NASA, Tech. Rep. CR-2014-218178, 2014.

[4] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd. Springer, 2000.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[6] S. Obayashi, Y. Yamaguchi, and T. Nakamura, "Multiobjective Genetic Algorithm for Multidisciplinary Design of Transonic Wing Planform," *Journal of Aircraft*, vol. 34, no. 5, pp. 690–693, 1997.

[7] H. Arizono and K. Isogai, "Application of Genetic Algorithm for Aeroelastic Tailoring of a Cranked-arrow Wing," *Journal of Aircraft*, vol. 42, no. 2, pp. 493–499, 2005.

[8] J. Kennedy, "Particle Swarm Optimization," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds. Boston, MA: Springer US, 2010, pp. 760–766, ISBN: 978-0-387-30164-8.

[9] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[10] P. J. M. van Laarhoven and E. H. L. Aarts, "Simulated Annealing," in *Simulated Annealing: Theory and Applications*. Dordrecht: Springer Netherlands, 1987, pp. 7–15, ISBN: 978-94-015-7744-1.

[11] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz, "BOA: The Bayesian Optimization Algorithm," in *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation - Volume 1*, Orlando, Florida: Morgan Kaufmann Publishers Inc., 1999, pp. 525–532.

[12] D. C. Karnopp, "Random Search Techniques for Optimization Problems," *Automatica*, vol. 1, no. 2, pp. 111 –121, 1963.

[13] J. Bergstra and Y. Bengio, "Random Search for Hyper-parameter Optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.

[14] L. Grippo, F. Lampariello, and S. Lucidi, "A Nonmonotone Line Search Technique for Newton's Method," *SIAM Journal on Numerical Analysis*, vol. 23, no. 4, pp. 707–716, 1986.

[15] W. W. Hager and H. Zhang, "A new Conjugate Gradient Method with Guaranteed Descent and an Efficient Line Search," *SIAM Journal on optimization*, vol. 16, no. 1, pp. 170–192, 2005.

[16] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*. MOS/SIAM, 2000.

[17] R. H. Byrd, J. C. Gilbert, and J. Nocedal, "A Trust Region Method based on Interior Point Techniques for Nonlinear Programming," *Mathematical Programming*, vol. 89, no. 1, pp. 149–185, 2000.

[18] Z. Lyu, Z. Xu, and J. Martins, "Benchmarking Optimization Algorithms for Wing Aerodynamic Design Optimization," in *Proceedings of the 8th International Conference on Computational Fluid Dynamics*, Chengdu, Sichuan, China, 2014.

[19] H. H. Rosenbrock, "An Automatic Method for Finding the Greatest or Least Value of a Function," *The Computer Journal*, vol. 3, no. 3, pp. 175–184, Jan. 1960.

[20] P. Gill, W. Murray, and M. Saunders, "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, vol. 47, no. 1, pp. 99–131, 2005.

[21] J. Lyness, "Numerical Algorithms based on the Theory of Complex Variable," in *Proceedings of the 1967 22nd national conference*, ACM, 1967, pp. 125–133.

[22] W. K. Anderson, J. C. Newman, D. L. Whitfield, and E. J. Nielsen, "Sensitivity Analysis for Navier-Stokes Equations on Unstructured Meshes Using Complex Variables," *AIAA Journal*, vol. 39, no. 1, pp. 56–63, 2001.

[23] J. Martins, P. Sturdza, and J. Alonso, "The Complex-step Derivative Approximation," *ACM Transactions on Mathematical Software (TOMS)*, vol. 29, no. 3, pp. 245–262, 2003.

[24] J. C. Newman, W. K. Anderson, and D. L. Whitfield, "Multidisciplinary Sensitivity Derivatives using Complex Variables," Mississippi State University Publication, Tech. Rep. MSSU-EIRS-ERC-98-08, 1998.

[25] A. Jameson, "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, vol. 3, no. 3, pp. 233–260, 1988.

[26] E. J. Nielsen and W. K. Anderson, "Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes," *AIAA Journal*, vol. 40, pp. 1155–1163, 2002.

[27] S. Nadarajah, A. Jameson, and J. Alonso, "Sonic Boom Reduction Using an Adjoint Method for Wing-Body Configurations in Supersonic Flow," in *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. American Institute of Aeronautics and Astronautics, 2002.

[28] B. Y. Zhou, T. A. Albring, N. R. Gauger, T. D. Economon, F. Palacios, and J. J. Alonso, "A Discrete Adjoint Framework for Unsteady Aerodynamic and Aeroacoustic Optimization," in *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. American Institute of Aeronautics and Astronautics, 2015.

[29] K. J. Fidkowski and D. L. Darmofal, "Review of Output-based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics," *AIAA Journal*, vol. 49, no. 4, pp. 673–694, 2011.

[30] T. Oliver and D. Darmofal, "Analysis of Dual Consistency for Discontinuous Galerkin Discretizations of Source Terms," *SIAM Journal on Numerical Analysis*, vol. 47, no. 5, pp. 3507–3525, 2009. eprint: `https://doi.org/10.1137/080721467`.

[31] J. E. Peter and R. P. Dwight, "Numerical Sensitivity Analysis for Aerodynamic Optimization: A Survey of Approaches," *Computers & Fluids*, vol. 39, no. 3, pp. 373 –391, 2010.

[32] J. Berg and J. Nordström, "On the Impact of Boundary Conditions on Dual Consistent Finite Difference Discretizations," *Journal of Computational Physics*, vol. 236, pp. 41 –55, 2013.

[33] A. C. Huang, S. R. Allmaras, M. C. Galbraith, and D. L. Darmofal, "Well-posed Subsonic Inflow/Outflow Boundary Conditions for the Navier-Stokes Equations," in *2018 AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, 2018.

[34] C. Bischof, C. Corliss, L. L. Green, A. Griewank, K. J. Haigler, and P. A. Newman, "Automatic Differentiation of Advanced CFD Codes for Multidisciplinary Design," Tech. Rep., 2003.

[35] D. A. Fournier, H. J. Skaug, J. Ancheta, J. Ianelli, A. Magnusson, M. N. Maunder, A. Nielsen, and J. Sibert, "AD Model Builder: using Automatic Differentiation for

Statistical Inference of Highly Parameterized Complex Nonlinear Models," *Optimization Methods and Software*, vol. 27, no. 2, pp. 233–249, 2012.

[36] J.-D. Müller and P Cusdin, "On the Performance of Discrete Adjoint CFD Codes using Automatic Differentiation," *International Journal for Numerical Methods in Fluids*, vol. 47, no. 8-9, pp. 939–945, 2005.

[37] T. A. Weisshaar, "Aeroelastic Optimization of a Panel in High Mach Number Supersonic Flow," *Journal of Aircraft*, vol. 9, no. 9, pp. 611–617, 1972.

[38] R. T. Haftka and J. H. S. Jr., "WINDOWAC (Wing Design Optimization with Aeroelastic Constraints): Program Manual," NASA, Tech. Rep. TM-X-3071, 1974.

[39] B. L. Pierson, "Aeroelastic Panel Optimization with Aerodynamic Damping," *AIAA Journal*, vol. 13, no. 4, pp. 515–517, 1975.

[40] A. P. Seyranian, "Sensitivity Analysis and Optimization of Aeroelastic Stability," *International Journal of Solids and Structures*, vol. 18, no. 9, pp. 791 –807, 1982.

[41] D. J. Neill, E. H. Johnson, and R. Canfield, "ASTROS - A Multidisciplinary Automated Structural Design Tool," *Journal of Aircraft*, vol. 27, no. 12, pp. 1021–1027, 1990.

[42] J. K. S. Dillinger, T. Klimmek, M. M. Abdalla, and Z. Gürdal, "Stiffness Optimization of Composite Wings with Aeroelastic Constraints," *Journal of Aircraft*, vol. 50, no. 4, pp. 1159–1168, 2013.

[43] B. K. Stanford and P. S. Beran, "Analytical Sensitivity Analysis of an Unsteady Vortex-Lattice Method for Flapping-Wing Optimization," *Journal of Aircraft*, vol. 47, no. 2, pp. 647–662, 2010.

[44] D. M. De Leon, C. E. de Souza, J. S. O. Fonseca, and R. G. A. da Silva, "Aeroelastic Tailoring using Fiber Orientation and Topology Optimization," *Structural and Multidisciplinary Optimization*, vol. 46, no. 5, pp. 663–677, 2012.

[45] M. H. Shirk, T. J. Hertz, and T. A. Weisshaar, "Aeroelastic Tailoring - Theory, Practice, and Promise," *Journal of Aircraft*, vol. 23, no. 1, pp. 6–18, 1986.

[46] E. Livne, "Integrated Aeroservoelastic Optimization: Status and Direction," *Journal of Aircraft*, vol. 36, no. 1, pp. 122–145, 1999.

[47] K Maute and M Allen, "Conceptual Design of Aeroelastic Structures by Topology Optimization," *Structural and Multidisciplinary Optimization*, vol. 27, no. 1-2, pp. 27–42, 2004.

[48] C. L. Pettit, "Uncertainty Quantification in Aeroelasticity: Recent Results and Research Challenges," *Journal of Aircraft*, vol. 41, no. 5, pp. 1217–1229, 2004.

[49] C. L. Pettit and R. V. Grandhi, "Optimization of a Wing Structure for Gust Response and Aileron Effectiveness," *Journal of Aircraft*, vol. 40, no. 6, pp. 1185–1191, 2003.

[50] B. Stanford, "Aeroservoelastic Optimization under Stochastic Gust Constraints," in *2018 Applied Aerodynamics Conference*. American Institute of Aeronautics and Astronautics, 2018.

[51] R. M. Kolonay and H. T. Y. Yang, "Unsteady Aeroelastic Optimization in the Transonic Regime," *Journal of Aircraft*, vol. 35, no. 1, pp. 60–68, 1998.

[52] D. J. Lucia, P. S. Beran, and W. A. Silva, "Reduced-order Modeling: New Approaches for Computational Physics," *Progress in Aerospace Sciences*, vol. 40, no. 1, pp. 51 –117, 2004.

[53] K. C. Hall, J. P. Thomas, and E. H. Dowell, "Proper Orthogonal Decomposition Technique for Transonic Unsteady Aerodynamic Flows," *AIAA Journal*, vol. 38, no. 10, pp. 1853–1862, 2000.

[54] K. Willcox and J. Peraire, "Balanced Model Reduction via the Proper Orthogonal Decomposition," *AIAA Journal*, vol. 40, no. 11, pp. 2323–2330, 2002.

[55] K.-S. Zhang, Z.-H. Han, W.-J. Li, and W.-P. Song, "Coupled Aerodynamic/Structural Optimization of a Subsonic Transport Wing Using a Surrogate Model," *Journal of Aircraft*, vol. 45, no. 6, pp. 2167–2171, 2008.

[56] R. Liem, G. Kenway, and J. Martins, "Multi-point, Multi-mission, High-fidelity Aerostructural Optimization of a Long-range Aircraft Configuration," in *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ ISSMO Multidisciplinary Analysis and Optimization Conference*, 2012, p. 5706.

[57] W. Zhiqiang, W. Xiaozhe, and Y. Chao, "A Highly Efficient Aeroelastic Optimization Method Based on a Surrogate Model," *International Journal of Aeronautical and Space Sciences*, vol. 17, no. 4, pp. 491–500, 2016.

[58] A. Mannarino and P. Mantegazza, "Nonlinear Aeroelastic Reduced Order Modeling by Recurrent Neural Networks," *Journal of Fluids and Structures*, vol. 48, pp. 103 –121, 2014.

[59] J. Reuther, J. Alonso, J. Martins, and S. Smith, "A Coupled Aero-structural Optimization Method for Complete Aircraft Configurations," in *37th Aerospace Sciences Meeting and Exhibit*, 1999, p. 187.

[60] K. Maute, M. Nikbay, and C. Farhat, "Coupled Analytical Sensitivity Analysis and Optimization of Three-Dimensional Nonlinear Aeroelastic Systems," *AIAA Journal*, vol. 39, no. 11, pp. 2051–2061, 2001.

[61] J. Martins, J. J. Alonso, and J. J. Reuther, "High-Fidelity Aerostructural Design Optimization of a Supersonic Business Jet," *Journal of Aircraft*, vol. 41, no. 3, pp. 523–530, 2004.

[62] J. Martins, J. Alonso, and J. Reuther, "A Coupled-Adjoint Sensitivity Analysis Method for High-Fidelity Aero-Structural Design," *Optimization and Engineering*, vol. 6, no. 1, pp. 33–62, 2005.

[63] G. J. Kennedy and J. Martins, "Parallel Solution Methods for Aerostructural Analysis and Design Optimization," in *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, AIAA 2010-9308, Fort Worth, TX, Jan. 1, 2010, published.

[64] J. Brezillon, A. Ronzheimer, D. Haar, M. Abu-Zurayk, M. Lummer, W. Kruger, and F. J. Natterer, "Development and Application of Multi-disciplinary Optimization Capabilities based on High-fidelity Methods," in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. American Institute of Aeronautics and Astronautics, 2012.

[65] I. Ghazlane, G. Carrier, A. Dumont, and J.-a. Desideri, "Aerostructural Adjoint Method for Flexible Wing Optimization," in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. American Institute of Aeronautics and Astronautics, 2012.

[66] G. Kenway, G. Kennedy, and J. Martins, "Aerostructural Optimization of the Common Research Model Configuration," in *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, American Institute of Aeronautics and Astronautics, 2014.

[67] G. K. W. Kenway and J. R.R. A. Martins, "Multipoint High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration," *Journal of Aircraft*, vol. 51, no. 1, pp. 144–160, 2014.

[68] J. Martins, G. Kennedy, and G. K. Kenway, "High Aspect Ratio Wing Design: Optimal Aerostructural Tradeoffs for the Next Generation of Materials," in *52nd Aerospace Sciences Meeting*, 2014, p. 0596.

[69] K. A. James, G. J. Kennedy, and J. R. Martins, "Concurrent Aerostructural Topology Optimization of a Wing Box," *Computers & Structures*, vol. 134, pp. 1–17, 2014.

[70] K. Palaniappan, P. Sahu, J. Alonso, and A. Jameson, "Active Flutter Control using an Adjoint Method," in *44th AIAA Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, 2006.

[71] Q. Wang, P. Moin, and G. Iaccarino, "Minimal Repetition Dynamic Checkpointing Algorithm for Unsteady Adjoint Calculation," *SIAM Journal on Scientific Computing*, vol. 31, no. 4, pp. 2549–2567, 2009.

[72] E. J. Nielsen, B. Diskin, and N. K. Yamaleev, "Discrete Adjoint-based Design Optimization of Unsteady Turbulent Flows on Dynamic Unstructured Grids," *AIAA Journal*, vol. 48, no. 6, pp. 1195–1206, 2010.

[73] E. J. Nielsen and B. Diskin, "Discrete Adjoint-based Design for Unsteady Turbulent Flows on Dynamic Overset Unstructured Grids," *AIAA Journal*, vol. 51, no. 6, pp. 1355–1373, 2013.

[74] R. Roth and S. Ulbrich, "A Discrete Adjoint Approach for the Optimization of Unsteady Turbulent Flows," *Flow, Turbulence and Combustion*, vol. 90, no. 4, pp. 763–783, 2013.

[75] T. Economon, F. Palacios, and J. Alonso, "Unsteady Aerodynamic Design on Unstructured Meshes with Sliding Interfaces," in *51st AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, p. 632.

[76] P. J. Blonigan, Q. Wang, E. J. Nielsen, and B. Diskin, "Least-Squares Shadowing Sensitivity Analysis of Chaotic Flow Around a Two-Dimensional Airfoil," *AIAA Journal*, vol. 56, no. 2, pp. 658–672, 2017.

[77] Q. Wang, "Forward and Adjoint Sensitivity Computation of Chaotic Dynamical Systems," *Journal of Computational Physics*, vol. 235, pp. 1–13, 2013.

[78] P. J. Blonigan and Q. Wang, "Multiple Shooting Shadowing for Sensitivity Analysis of Chaotic Dynamical Systems," *Journal of Computational Physics*, vol. 354, pp. 447–475, 2018.

[79] K. Mani and D. J. Mavriplis, "Adjoint-Based Sensitivity Formulation for Fully Coupled Unsteady Aeroelasticity Problems," *AIAA Journal*, vol. 47, no. 8, pp. 1902–1915, 2009.

[80] A. Mishra, D. Mavriplis, and J. Sitaraman, "Time-Dependent Aeroelastic Adjoint-Based Aerodynamic Shape Optimization of Helicopter Rotors in Forward Flight," *AIAA Journal*, pp. 3813–3827, 2016.

[81] E. Fabiano, "Multidisciplinary Adjoint-based Design Optimization Techniques for Helicopter Rotors," PhD thesis, University of Wyoming, Laramie, Wyoming, 2017.

[82] E. Fabiano and D. J. Mavriplis, "Adjoint-based Aeroacoustic Design-Optimization of Flexible Rotors in Forward Flight," in *AHS 72nd Annual Forum*, West Palm Beach, Florida, 2016.

[83] E. Fabiano and D. Mavriplis, "Adjoint-based Aeroacoustic Design Optimization of Flexible Rotors in Forward Flight," *Journal of the American Helicopter Society*, vol. 62, no. 4, pp. 1–17, 2017.

[84] J. E. F. Williams and D. L. Hawkings, "Sound Generation by Turbulence and Surfaces in Arbitrary Motion," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 264, no. 1151, pp. 321–342, 1969.

[85] L. Wang, B. Diskin, R. Biedron, E. J. Nielsen, and O. Bauchau, "Sensitivity Analysis of Multidisciplinary Rotorcraft Simulations," in *55th AIAA Aerospace Sciences Meeting*, 2017, p. 1670.

[86] O. A. Bauchau, "DYMORE User's manual," *Georgia Institute of Technology, Atlanta*, 2007.

[87] *FUN3D 13.3 Manual*, 2018.

[88] A. Callejo, O. Bauchau, B. Diskin, and L. Wang, "Sensitivity Analysis of Beam Cross-Section Stiffness Using Adjoint Method," no. 58202, V006T10A058–, 2017.

[89] Z. Zhang, P. C. Chen, S. Yang, Z. Wang, and Q. Wang, "Unsteady Aerostructure Coupled Adjoint Method for Flutter Suppression," *AIAA Journal*, vol. 53, no. 8, pp. 2121–2129, 2015.

[90] Z. Zhang, P.-C. Chen, Q. Wang, Z. Zhou, S. Yang, and Z. Wang, "Adjoint-based Structure and Shape Optimization with Flutter Constraints," in *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, 2016.

[91] Z. Zhang, Q. Wang, P.-C. Chen, S. Yang, and Z. Wang, "CFD-based Aeroelastic Adjoint Sensitivities for Flight Vehicle Weight Minimization Using Sizing and Shape Design Variables," in *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, 2017.

[92] M. Widhalm, R. P. Dwight, R. Thormann, and A Hübner, "Efficient Computation of Dynamic Stability Data with a Linearized Frequency Domain Solver," in *Proceedings of the European Conference on Computational Fluid Dynamics*, Lisbon, Portugal, 2010.

[93] K. C. Hall, J. P. Thomas, and W. S. Clark, "Computation of Unsteady Nonlinear Flows in Cascades Using a Harmonic Balance Technique," *AIAA Journal*, vol. 40, no. 5, pp. 879–886, 2002.

[94] J. P. Thomas, E. H. Dowell, and K. C. Hall, "Modeling Viscous Transonic Limit Cycle Oscillation Behavior Using a Harmonic Balance Approach," *Journal of Aircraft*, vol. 41, no. 6, pp. 1266–1274, 2004.

[95] R. Thormann and S. Timme, "Application of Harmonic Balance Method for Nonlinear Gust Responses," in *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, 2018.

[96] P. Bekemeyer, R. Thormann, and S. Timme, "Frequency-Domain Gust Response Simulation Using Computational Fluid Dynamics," *AIAA Journal*, vol. 55, no. 7, pp. 2174–2185, 2017.

[97] W. K. Anderson and D. L. Bonhaus, "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers & Fluids*, vol. 23, no. 1, pp. 1–21, 1994.

[98] A. J. Chorin, "A Numerical Method for Solving Incompressible Viscous Flow Problems," *Journal of Computational Physics*, vol. 135, no. 2, pp. 118–125, 1997.

[99] R. T. Biedron and J. L. Thomas, "Recent Enhancements to the FUN3D Flow Solver for Moving-Mesh Applications," in *AIAA 47th Aerospace Sciences Meeting*, Orlando, Florida, 2009.

[100] P Spalart and S Allmaras, "A One-equation Turbulence Model for Aerodynamic Flows," in *30th Aerospace Sciences Meeting and Exhibit*, 1992, p. 439.

[101] F Menter, "Zonal Two Equation kw Turbulence Models for Aerodynamic Flows," in *23rd Fluid Dynamics, Plasmadynamics, and Lasers Conference*, 1993, p. 2906.

[102] P Spalart, W Jou, M Strelets, and S Allmaras, "Comments on the Feasibility of LES for Wings, and on a Hybrid RANS/LES Approach," *Advances in DNS/LES*, vol. 1, pp. 4–8, 1997.

[103] P. Spalart, S. Deck, M Shur, K Squires, M. K. Strelets, and A Travin, "A new version of Detached-eddy Simulation, Resistant to Ambiguous Grid Densities," *Theoretical and Computational Fluid Dynamics*, vol. 20, no. 3, pp. 181–195, 2006.

[104] M. Sanchez-Rocha, M. Kirtas, and S. Menon, "Zonal Hybrid RANS-LES Method for Static and Oscillating Airfoils and Wings," in *44th AIAA Aerospace Sciences Meeting and Exhibit*, 2006, p. 1256.

[105] J. Smagorinsky, "General Circulation Experiments with the Primitive Equations: I. The Basic Experiment," *Monthly Weather Review*, vol. 91, no. 3, pp. 99–164, 1963.

[106] W. Johnson, "Technology Drivers in the Development of CAMRAD II," in *American Helicopter Society Aeromechanics Specialist Conference, San Fransisco, CA, January*, 1994, pp. 19–21.

[107] E. J. Nielsen, J. Lu, M. A. Park, and D. L. Darmofal, "An Implicit, Exact Dual Adjoint Solution Method for Turbulent Flows on Unstructured Grids," *Computers & Fluids*, vol. 33, no. 9, pp. 1131–1155, 2004.

[108] M. A. Park, "Adjoint-based, Three-dimensional Error Prediction and Grid Adaptation," *AIAA journal*, vol. 42, no. 9, pp. 1854–1862, 2004.

[109] E. J. Nielsen and M. A. Park, "Using an Adjoint Approach to Eliminate Mesh Sensitivities in Computational Design," *AIAA Journal*, vol. 44, no. 5, pp. 948–953, 2006.

[110] E. J. Nielsen, E. M. Lee-Rausch, and W. T. Jones, "Adjoint-based Design of Rotors in a Noninertial Reference Frame," *Journal of Aircraft*, vol. 47, no. 2, pp. 638–646, 2010.

[111] G. J. Kennedy and J. Martins, "A Parallel Finite-element Framework for Large-scale Gradient-based Design Optimization of High-performance Structures," *Finite Elements in Analysis and Design*, vol. 87, pp. 56–73, 2014.

[112] K. Boopathy and G. Kennedy, "Adjoint-based Derivative Evaluation Methods for Flexible Multibody Systems with Rotorcraft Applications," in *55th AIAA Aerospace Sciences Meeting*, 2017.

[113] T. R. Brooks, G. Kennedy, and J. Martins, "High-fidelity Aerostructural Optimization of a High Aspect Ratio Tow-steered Wing," in *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2016, p. 1179.

[114] J. F. Kiviaho, K. E. Jacobson, M. J. Smith, and G. J. Kennedy, "A Robust and Flexible Coupling Framework for Aeroelastic Analysis and Optimization," in *Proceedings of the 19th AIAA/ISSMO Multidisciplinary Analysis and Optmization Conference*, AIAA, Denver, Colorado, 2017.

[115] K. E. Jacobson, J. F. Kiviaho, M. J. Smith, and G. J. Kennedy, "An Aeroelastic Coupling Framework for Time-Accurate Analysis and Optimization," in *Proceedings of the 56th AIAA Aerospace Sciences Meeting*, AIAA, Kissimmee, Florida, 2018.

[116] J. F. Kiviaho and G. J. Kennedy, "Robust Load and Displacement Transfer Schemes using a Weighted Least-squares Method," 2017, Submitted.

[117] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares Fitting of Two 3-D Point Sets," *IEEE Transactions on pattern analysis and machine intelligence*, no. 5, pp. 698–700, 1987.

[118] C. Farhat, M. Lesoinne, and P. LeTallec, "A Conservative Algorithm for Exchanging Aerodynamic and Elastodynamic Data in Aeroelastic Systems," in *36th AIAA Aerospace Sciences Meeting and Exhibit*, 1998, p. 515.

[119] M. J. Smith, "Conservation Issues for RANS-Based Rotor Aeroelastic Simulations," *Journal of Aerospace Engineering*, vol. 25, no. 2, pp. 217–228, 2012.

[120] B. K. Stanford and P. D. Dunning, "Optimal Topology of Aircraft Rib and Spar Structures under Aeroelastic Loads," *Journal of Aircraft*, vol. 52, no. 4, pp. 1298–1311, 2014.

[121] J. Samareh, "Multidisciplinary Aerodynamic-structural Shape Optimization using Deformation (MASSOUD)," in *8th Symposium on Multidisciplinary Analysis and Optimization*. American Institute of Aeronautics and Astronautics, 2000.

[122] A. Watt and M. Watt, *Advanced Animation and Rendering Techniques*. Addison-Wesley, 1992.

[123] R. D. Snyder, "A Cross-Language Remote Procedure Call Framework," in *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. American Institute of Aeronautics and Astronautics, 2017.

[124] P. Hintjens, *ZeroMQ: Messaging for many Applications*. " O'Reilly Media, Inc.", 2013.

[125] B. M. Irons and R. C. Tuck, "A Version of the Aitken Accelerator for Computer Iteration," *International Journal for Numerical Methods in Engineering*, vol. 1, no. 3, pp. 275–277, 1969.

[126] W. Karush, "Minima of Functions of Several Variables with Inequalities as Side Conditions," PhD thesis, 1939.

[127] M. A. Hanson, "On Sufficiency of the Kuhn-Tucker Conditions," *Journal of Mathematical Analysis and Applications*, vol. 80, no. 2, pp. 545–550, 1981.

[128] J. W. Banks, W. D. Henshaw, A. K. Kapila, and D. W. Schwendeman, "An Added-mass Partition Algorithm for Fluid-structure Interactions of Compressible Fluids

and Nonlinear Solids," *Journal of Computational Physics*, vol. 305, pp. 1037–1064, 2016.

[129] P. Anagnostopoulos and P. Bearman, "Response Characteristics of a Vortex-excited Cylinder at Low Reynolds Numbers," *Journal of Fluids and Structures*, vol. 6, no. 1, pp. 39 –50, 1992.

[130] J. Heeg, P. Chwalowski, D. Schuster, and M. Dalenbring, "Overview and Lessons Learned from the Aeroelastic Prediction Workshop," in *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, 2013.

[131] J. Heeg, C. Wieseman, and P. Chwalowski, "Overview and Data Comparisons from the 2nd Aeroelastic Prediction Workshop," in *34th AIAA Applied Aerodynamics Conference*. American Institute of Aeronautics and Astronautics, 2016.

[132] P. Chwalowski and J. Heeg, "FUN3D Analyses in Support of the Second Aeroelastic Prediction Workshop," in *34th AIAA Applied Aerodynamics Conference*. American Institute of Aeronautics and Astronautics, 2016.

[133] "Aeroelastic Prediction Workshop II, https://nescacademy.nasa.gov/workshops/AePW2/public/,"

[134] P. Chwalowski, J. Heeg, and R. Biedron, "Numerical Investigations of the Benchmark Supercritical Wing in Transonic Flow," in *55th AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, 2017.

[135] G. Kreisselmeier and R. Steinhauser, "Systematic Control Design by Optimizing a Vector Performance Index," in *International Federation of Active Controls Symposium on Computer-Aided Design of Control Systems*, Zurich, Switzerland, 1979.

[136] G. J. Kennedy, "Strategies for Adaptive Optimization with Aggregation Constraints using Interior-point Methods," *Computers & Structures*, vol. 153, pp. 217–229, Mar. 1, 2015, published.

[137] G. J. Kennedy and J. E. Hicken, "Improved Constraint-Aggregation Methods," *Computer Methods in Applied Mechanics and Engineering*, vol. 289, pp. 332–354, Mar. 1, 2015, published.

[138] M. W. Kehoe, "A Historical Overview of Flight Flutter Testing," Tech. Rep., 1995.

[139] X. Fang and H. Luo, "An Improved Hilbert Transform for Nonlinear Vibration Signal Analysis," in *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, 2009.

[140] N. K. Mandal, R. A. Rahman, and M. S. Leong, "Experimental Study on Loss Factor for Corrugated Plates by Bandwidth Method," *Ocean Engineering*, vol. 31, no. 10, pp. 1313–1323, 2004.

[141] S. A. Badsar, M. Schevenels, W. Haegeman, and G. Degrande, "Determination of the Material Damping Ratio in the Soil from SASW Tests using the Half-power Bandwidth Method," *Geophysical Journal International*, vol. 182, no. 3, pp. 1493–1508, Sep. 2010.

[142] G. A. Papagiannopoulos and G. D. Hatzigeorgiou, "On the use of the Half-power Bandwidth Method to Estimate Damping in Building Structures," *Soil Dynamics and Earthquake Engineering*, vol. 31, no. 7, pp. 1075–1079, 2011.

[143] Y. Hua and T. K. Sarkar, "Matrix Pencil Method for Estimating Parameters of Exponentially Damped/Undamped Sinusoids in Noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 5, pp. 814–824, 1990.

[144] J. Razavilar, Y. Li, and K. J. R. Liu, "Spectral Estimation based on Structured Low Rank Matrix Pencil," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 5, 1996, 2503–2506 vol. 5.

[145] T. K. Sarkar and O. Pereira, "Using the Matrix Pencil Method to Estimate the Parameters of a Sum of Complex Exponentials," *IEEE Antennas and Propagation Magazine*, vol. 37, no. 1, pp. 48–55, 1995.

[146] J. Heeg, B. Stanford, C. D. Wieseman, S. Massey, J. Moore, and R. A. Truax, "Status Report on Aeroelasticity in the Vehicle Development for X-57 Maxwell," in *2018 Applied Aerodynamics Conference*. American Institute of Aeronautics and Astronautics, 2018.

[147] D. Kraft, "A Software package for Sequential Quadratic Programming," Köln, Germany, Tech. Rep. DFVLR-FB 88-28, 1988.

[148] R. E. Perez, P. W. Jansen, and J. R.R. A. Martins, "pyOpt: a Python-based Object-oriented Framework for Nonlinear Constrained Optimization," *Structural and Multidisciplinary Optimization*, vol. 45, no. 1, pp. 101–118, 2012.

[149] R. L. Bisplinghoff, H. Ashley, and R. L. Halfman, *Aeroelasticity*. Courier Corporation, 2013.

[150] J. D. Anderson Jr, *Fundamentals of Aerodynamics*. Tata McGraw-Hill Education, 2010.

[151] C. E. Y. Jr., "AGARD Standard Aeroelastic Configurations for Dynamic Response. Candidate Configuration I - Wing 445.6," NASA, Tech. Rep. TM-100492, 1987.

[152] W. A. Silva, P. Chwalowski, and B. Perry, "Evaluation of Linear, Inviscid, Viscous, and Reduced-order Modelling Aeroelastic Solutions of the AGARD 445.6 Wing using Root Locus Analysis," *International Journal of Computational Fluid Dynamics*, vol. 28, no. 3-4, pp. 122–139, Mar. 2014.

[153] FAA, "Dynamic Gust Loads," Tech. Rep. AC-23.341-1, 2014.

[154] D. E. Raveh, "CFD-Based Models of Aerodynamic Gust Response," *Journal of Aircraft*, vol. 44, no. 3, pp. 888–897, 2007.

[155] R. E. Bartels, "Development, Verification and Use of Gust Modeling in the NASA Computational Fluid Dynamics Code FUN3D," Tech. Rep., 2012.

[156] J. Vassberg, M. Dehaan, M. Rivers, and R. Wahls, "Development of a Common Research Model for Applied CFD Validation Studies," in *26th AIAA Applied Aerodynamics Conference*. American Institute of Aeronautics and Astronautics, 2008.

[157] E. N. Tinoco, O. Brodersen, S. Keye, and K. Laflin, "Summary of Data from the Sixth AIAA CFD Drag Prediction Workshop: CRM Cases 2 to 5," in *55th AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, 2017.

[158] C. L. Rumsey, J. P. Slotnick, and A. J. Sclafani, "Overview and Summary of the Third AIAA High Lift Prediction Workshop," in *2018 AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, 2018.

[159] K.-J. Bathe and E. N. Dvorkin, "A Formulation of General Shell Elements–the use of Mixed Interpolation of Tensorial Components," *International Journal for Numerical Methods in Engineering*, vol. 22, no. 3, pp. 697–722, 1986.

[160] A. Bowers, O. Murillo, R. Jensen, B. Eslinger, and C. Gelzer, "On Wings of the Minimum Induced Drag: Spanload Implications for Aircraft and Birds," NASA, Tech. Rep. NASA/TP-2016-219072, 2016.

[161] R. Haimes and M. Drela, "On The Construction of Aircraft Conceptual Geometry for High-Fidelity Analysis and Design," in *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics, 2012.

[162] G. Kennedy, "Adjoint-based Mesh Adaptation Techniques for Aggregation Functionals in Structural Design Optimization," in *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2016, p. 4127.